Finite element analysis of coupled electromechanical problems

Melgoza-Vazquez, Enrique

Award date:
2001

Awarding institution:
University of Bath

Finite Element Analysis of Coupled Electromechanical Problems

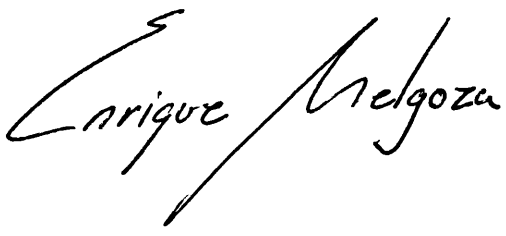
Submitted by Enrique Melgoza-Vázquez

for the Degree of Doctor of Philosophy

of the University of Bath

2001

COPYRIGHT



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

The modeling of electromechanical problems is discussed. The simultaneous consideration of two distinct phenomena is required, as the evolution of the electromagnetic and the mechanical parts are influenced by each other. In this work the equations of the coupled problem are described and possible methods of solution are considered.

Three general approaches with varying degrees of detail are considered. In the first, a lumped parameter model of the device is constructed from the finite element solution of the electromagnetic problem. A second approach links the electromagnetic field directly with the lumped mechanical part. Lastly, both the electromagnetic and the mechanical systems are considered to be distributed, with the individual domains solved by using the finite element method.

In the process of solution of transient problems the need to solve differential-algebraic systems of equations arises and some approaches are presented. It is shown that traditional finite difference formulas may be applied as long as the discretization is made at the element level. Higher order methods and step adaptation are discussed.

Acknowledgments

I would like to thank my wife Janeth and my son Andrés, and the rest of our families for all their support and encouragement.

Thanks to Prof.D.Rodger, Dr.H.C.Lai, Dr.P.C.Coles and Mr.R.Hill-Cottingham for their valuable suggestions and for providing the groundwork from where this thesis developed.

I acknowledge the financial support of Conacyt and SEP during the past years and thank Instituto Tecnológico de Morelia, Programa de Graduados e Investigación en Ingeniería Eléctrica, and Dr.J.L.Guardado for their support.

Contents

Chapter 1

Introduction

Electromechanical energy conversion devices play a very important role in almost any area of human endeavour in a modern society. The reason for this is that they provide a very convenient form of directing, controlling or applying force or energy. They are to be found in such dissimilar applications and scales as microelectromechanical sensors and actuators to gigawatt generators, and new application areas are constantly being added to this spectrum.

Analysis methods for electromechanical devices have thus been constantly pushed to deliver increased fidelity. The wide availability of computers has transformed the way in which analysis and design are carried out, and analysis techniques are now at a stage where the simultaneous consideration of interactions from different physical fields is being investigated.

Applications of coupled phenomena include, besides the broad area of electrical machines, electrostrictive and magnetostrictive systems [1, 2, 3], eddy current brakes [4, 5], magnetic levitation, magnetic forming, superconducting systems, etc.[6, 7]. The relatively new technology of MEMS (micro-electromechanical systems) relies,

as its name implies, on electromechanical effects, among others [8, 9, 10].

Novel applications for electromechanical systems are an active area of research. The motivation is the demand for high performance controlled actuators (featuring miniaturization, increased speed and higher accuracy) [11, 12, 13]. Rapid development of high-technology and applied physical sciences is requiring interdisciplinary research in many areas.

The finite element method provides a convenient framework for the determination of field quantities. The calculation of the magnetic field in energy conversion devices by using this technique is well documented. The consideration of the effect of the mechanical load is a recent research line.

Previous work in the subject is based almost entirely on the weak coupling of magnetic and mechanical systems[14, 15, 16, 17, 18, 19, 20, 21]. A strong coupling approach was presented by Ren[22].

The application of a distributed parameter mechanical code (elasticity) to a coupled problem between electromagnetic mechanical systems is considered in [23, 24, 25, 26, 27, 28, 29]

In this work, methods of analysis of electromechanical conversion devices are discussed. The finite element method is used to describe the electromagnetic part and the coupling with the mechanical side is made along three broad lines. In the first of them, the whole system (electromagnetic and mechanical) is reduced to a set of lumped parameters recorded in the form of tables; this corresponds to the classical energy conversion models based on energy principles and as such is better suited to cases with only a few coils and where the effect of the eddy currents is small. A

second approach is to connect the electromagnetic field model discretization to a rigid body description of the mechanical part; this alleviates some of the drawbacks of the reduction approach and now a full consideration of the eddy current phenomenon is possible. Finally, both the electromagnetic and the displacement fields are considered to be distributed.

Both static and dynamic problems are considered. In the case of dynamic problems, the time stepping scheme plays a crucial role and for this reason a separate chapter is devoted to the discussion of the different schemes available. Some other considerations regarding the time stepping of differential-algebraic systems of equations are also discussed.

The aim of this work is to propose and implement solution schemes that are stable and provide an accurate solution without demanding excessive computational resources.

Chapter 2

Reduced order models from finite element solutions

> *"Thus the properties of a coupling system can be determined completely if the electrical terminal relations are known and the system is represented by a conservative model"* [30], pp.70..

2.1 Introduction

Numerical solutions of field problems (using finite elements, finite differences, etc.) involving magnetic devices provide detailed and accurate performance predictions. However, they have some drawbacks: they are computationally intensive and are difficult to integrate into system-level simulation tools (circuit simulators or control system simulators). The field model requires a large number of equations to be solved. Higher level models are required if available tools for simulation are to be used.

On the other hand, traditional lumped parameter descriptions are fast and easy to incorporate into system models. These circuit models of magnetic devices evolved very early and preceded field models. Their disadvantage is that they are less accurate

since most of the time they are obtained by making idealizations and simplifications of both the material behavior and the geometry of the device.

It would be helpful if we can somehow have a model of the device which combines the accuracy of the large finite element models, with the convenience of the lumped models. Several approaches aiming at this goal have been proposed, among them: 1)the use of the finite element method to reproduce circuit measurements of the parameters, 2)modification of the finite element matrix by linear algebra and 3)tabulation of energy-related quantities calculated by several static finite element solutions. These alternatives will be briefly discussed in turn.

The first one is based on the reduction of the whole device to a network formed by discrete components (resistors, inductors and capacitors), one example of which is the equivalent circuit of an induction motor. The determination of the values of the parameters can be made by direct measurement, in some cases following standard procedures. The choice of topology of the equivalent circuit is guided by the observed behavior, but is to some extent arbitrary. The parameter determination can thus be seen as an attempt to produce the best fit, one that shows a good agreement between measured and calculated responses. This approach is appropriate to model static devices or dynamic devices in steady state. Finite element models provide a convenient means of determining parameters directly [31]. Frequently, mainly in applications related to control systems, a parameter identification procedure is performed by direct measurements [32, 33]. This is conceptually equivalent to the equivalent circuit determination, and the parameter identification could as well be based on finite element solutions instead [34].

The second general approach for model reduction consists in the manipulation of the finite element matrix [35, 36, 37]. The variables of the resulting reduced order

model may or may not represent physical quantities.

The third common lumped-parameter approach to model magnetic (or electric) devices is to use an energy method. The energy contained in the system is expressed in terms of a finite set of terminal variables (that is, variables which can be measured at ports) called the state variables. An analytical expression for the energy is favoured, but this is clearly only achievable for simple cases. Most of the time, the nonlinearities in the material and the complexity of the geometrical shapes make the determination of an analytical expression impossible. However, other methods for expressing the relation between the energy and the state variables are possible.

In this chapter we use the energy framework to obtain a reduced order model (one with a small number of equations) from a finite element model of the device. The new lumped variables will have physical significance, but are not necessarily circuit variables. We are interested in a particular case of reduction based on tables. In this, a look-up table together with some method of interpolation is used to specify the functional dependence between the state variables and the energy function. The first report essentially containing this approach is [38]. Other workers have elaborated on the method, almost all of them concentrating on a particular formulation detailed below. Recently, a new approach based on one table only has been proposed. In the following, after a review of the energy formulation, two new methods are proposed for the particular case of devices with one coil only. Devices with multiple coils are investigated thereafter.

Magnetic actuators of the solenoid valve type are used as examples, since they are particularly amenable to the analysis method proposed here. The subject is of interest in a number of applications; for instance, solenoid valves are used to control fuel injection in internal combustion engines [39]. Another relatively new area

of application is the mechanical driving of the contacts in power circuit breakers; the traditional spring mechanisms are replaced by a magnetic actuator, reducing maintenance requirements [40].

Despite new applications, the solenoid in itself has changed little since its inception. A textbook discussing this type of actuator, produced sixty years ago, asserted that the exact solution of the magnetic field and eddy current phenomena in moving configurations is not possible [41]. Fortunately, numerical approximations provide valuable information. Devices with axisymmetric geometry were first analyzed using the finite difference scheme in the 1980's [42]. Inclusion of the external circuit followed [15, 43]. Still today, efficient solutions for this case are being sought [44, 45], but it can be said that the problem can now be solved with the help of available tools, considering simultaneously the movement of the plunger, the connection of the external circuit and the eddy currents in the iron parts.

2.2 Energy methods

The theory of lumped-parameter models of electromechanical energy conversion devices is treated in a number of texts [46, 47, 30]. It is based on the notion of a lossless energy conversion mechanism, and briefly summarized here. We assume a one-coil device to simplify the notation.

The energy flow in a system with one electrical and one mechanical degrees of freedom obeys the relation

$$dW_m = id\lambda - f_e dx, \tag{2.1}$$

where W_m is the energy stored in the system (in the form of a magnetic field), i is the electric current, λ represents the flux linkage, f_e is the force of electromagnetic origin and x is the displacement. This expression is valid for conservative systems (those without dissipation). Put in words, (2.1) states that the energy injected into the system by electrical means is stored in the magnetic field if not converted into mechanical work.

2.2.1 λ and x as independent variables

The form of equation (2.1) suggests that λ and x can be considered the independent variables. In effect, if we regard the magnetic energy as a function of λ and x only, then:

$$W_m = W_m(\lambda, x), \tag{2.2}$$

and we can immediately write

$$dW_m = \frac{\partial W_m}{\partial \lambda} d\lambda + \frac{\partial W_m}{\partial x} dx \tag{2.3}$$

from where, upon comparison with (2.1),

$$i = \frac{\partial W_m}{\partial \lambda}, \tag{2.4}$$

$$f_e = -\frac{\partial W_m}{\partial x}. \tag{2.5}$$

2.2.2 i and x as independent variables

A Legendre transformation allows the use of an alternate set of independent variables. Knowing that

$$d(\lambda i) = \lambda di + i d\lambda, \tag{2.6}$$

the $id\lambda$ term in (2.1) is rewritten according to (2.6) to obtain

$$d(\lambda i - W_m) = \lambda di + f_e dx = dW'_m, \tag{2.7}$$

where the new variable W'_m (coenergy) has been defined. If we assume

$$W'_m = W'_m(i, x), \tag{2.8}$$

then

$$dW'_m = \frac{\partial W'_m}{\partial i} di + \frac{\partial W'_m}{\partial x} dx \tag{2.9}$$

and comparing (2.7) and (2.9)

$$\lambda = \frac{\partial W'_m}{\partial i}, \tag{2.10}$$

$$f_e = \frac{\partial W'_m}{\partial x}. \tag{2.11}$$

From (2.7), the energy and coenergy are related by

$$W_m + W'_m = \lambda i \tag{2.12}$$

2.2.3 State variables

Electrical and mechanical state variables are required for the simulation of the actuator. For the present case there is only one mechanical port so that the mechanical state equations are always the same, given by

$$\dot{v} = \frac{1}{M}[f_e - Bv - K(x - x_0)], \tag{2.13}$$

$$\dot{x} = v, \tag{2.14}$$

where v is the velocity, x the position, M the mass, B is the damping coefficient, K is the spring constant and x_0 the spring neutral position. The expressions for the electrical state variables and electromagnetic force f_e depend on the method used. Some possibilities are discussed in the following.

2.3 Table methods for one-coil devices

Equations (2.4),(2.5) and (2.10),(2.11) are the starting points for constructing a model based on tables. Several approaches are possible, but in practice the selection of a particular method will be determined by the results available from the FE computation. For instance, if the finite element program is capable of calculating the force, then a function $F_e(i,x)$ can be constructed. But if the force is not available then another method must be used. FE programs calculate the magnetic vector potential from a given current density distribution; this means that the (i,x) formulation is a natural way of developing the lumped-parameter model. However, (λ,x) formulations are also possible, as we will see. A plane 2D, axisymmetrical or 3D FE model may be used, depending on the particular problem at hand.

2.3.1 $(\mathbf{A}, F_e)$ Model

As mentioned above, the FE program most likely will require the specification of the current density, from which the magnetic vector potential $\mathbf{A}$ will be calculated. If the force is available from the FE program, then a series of calculations for a grid of current values and positions will provide a pair of tables

$$\mathbf{A} = \mathbf{A}(i,x), \tag{2.15}$$

$$F = F(i,x). \tag{2.16}$$

The flux linkage can be computed from (2.15) by using

$$\lambda = N \int_S \mathbf{B} \cdot d\mathbf{S} = N \int_S \nabla \times \mathbf{A} \cdot d\mathbf{S} = N \oint_L \mathbf{A} \cdot d\mathbf{L}, \tag{2.17}$$

where N is the number of turns in the coil. In the case of 2D models, this reduces to

$$\lambda = Nl(A_{in} - A_{out}), \tag{2.18}$$

where l is the length of the device perpendicular to the 2D plane and A_{in}, A_{out} are the values of the magnetic vector potential for the respective sides of the coil. As a result of these calculations, a table of flux linkage $\lambda(i, x)$ will be available. The electrical circuit equation is

$$V = Ri + \frac{d\lambda}{dt} = Ri + \frac{\partial \lambda}{\partial i}\frac{di}{dt} + v\frac{\partial \lambda}{\partial x}, \tag{2.19}$$

where V is the applied voltage and R the electrical resistance. From (2.19), the electrical state variable is the current, given by

$$\frac{di}{dt} = \frac{1}{\frac{\partial \lambda}{\partial i}}\left(V - Ri - v\frac{\partial \lambda}{\partial x}\right). \tag{2.20}$$

This method has been used by [38, 48, 49, 50, 51]. The coil will contain several nodes of the FE model, so that the location of the sampling point from which $\mathbf{A}(i, x)$ is taken is not unique (the values differ little but are not uniform over the region of the coil). Some kind of average process is desirable, but this is cumbersome to implement; alternatively, the value at a given node placed for instance at the center of the coil may be used. The latter approach is followed here since it generally gives good results.

2.3.2 (W_m, W'_m) Model

If both the energy and coenergy can be calculated, then the flux linkage is obtained from (2.12) since the current is known; the electrical state variable is the current and is again given by (2.20). On the other hand, the force is now calculated by using (2.11). This approach eliminates the randomness of choosing a special node for sampling the magnetic vector potential. The value so calculated is an averaged value since it is calculated from integral expressions. This method has not been reported before.

2.3.3 (W'_m) Model

It is possible to construct a model from only the knowledge of the coenergy. In this case, the flux linkage has to be computed from (2.10). Substituting in the circuit equation (2.19):

$$V = Ri + \frac{d\lambda}{dt} = Ri + \frac{\partial^2 W'_m}{\partial i^2}\frac{di}{dt} + \frac{\partial^2 W'_m}{\partial i \partial x}v. \tag{2.21}$$

The state equation is then

$$\frac{di}{dt} = \frac{1}{\frac{\partial^2 W'_m}{\partial i^2}}\left(V - Ri - v\frac{\partial^2 W'_m}{\partial i \partial x}\right) \tag{2.22}$$

The force is calculated directly by using (2.11). This approach has been proposed by [52, 53].

2.3.4 (A) Model

The three preceding models use the (i, x) formulation of the energy conversion process. It is also possible to use the (λ, x) formulation, but some modifications to the table data from the finite element solution are needed. If the magnetic vector potential $\mathbf{A}(i, x)$ is available, then it is possible to find an inverse function $i(\lambda, x)$. This can be seen in Fig.2.1, where the calculated flux linkage for a prototype actuator has been plotted against the current. The linkage-current relation is linear for large air gaps, but it gradually transforms into a curve following the nonlinear magnetization characteristic of the material for small air gaps. In any case, the relation is one to one and thus an inverse function $i(\lambda, x)$ can be calculated.

Once the inverse function $i(\lambda, x)$ has been found, the energy can be calculated by using

$$W_m = \int i d\lambda, \tag{2.23}$$

according to (2.4).

The electric state variable is λ in this case, the circuit equation being

$$V = Ri + \frac{d\lambda}{dt} \tag{2.24}$$

and the state equation

$$\frac{d\lambda}{dt} = V - Ri. \tag{2.25}$$

In (2.25), the current i is evaluated from the previously calculated function $i(\lambda, x)$, or

it may be approximated from (2.4), in which case a single table is required since the force is given by (2.5). This method, which has not been presented before, has the advantage of requiring only the magnetic vector potential solution, so that a generic package may be used, not necessarily designed for magnetic field computation, as long as the correct differential equation is specified.

A summary of the discussed possibilities is presented in Table 2.1. In the second method, (W_m, W_m), we could as well calculate the force and end up with the set (λ, F_e) as the basis for the simulation, just as in the $(\mathbf{A}, F_e)$ case. However, we will keep the method as stated, in order to verify new characteristics.

2.3.5 Inductance

Some authors introduce an inductance table instead of using directly the flux linkage [38, 54, 55, 56]. If the apparent inductance is defined as

$$L(i,x) = \frac{\lambda(i,x)}{i}, \tag{2.26}$$

then the circuit equation (2.19) leads to

$$\frac{di}{dt} = \frac{1}{L + i\frac{\partial L}{\partial i}} \left[V - \left(R + v\frac{\partial L}{\partial x} \right) i \right]. \tag{2.27}$$

When used together with a force table, this method is equivalent to $(\mathbf{A}, F)$. Very often, the $i\frac{\partial L}{\partial i}$ term in the denominator of (2.27) is omitted, but this is only valid for large air gaps, where the inductance may be regarded as a function only of the position, $L = L(x)$.

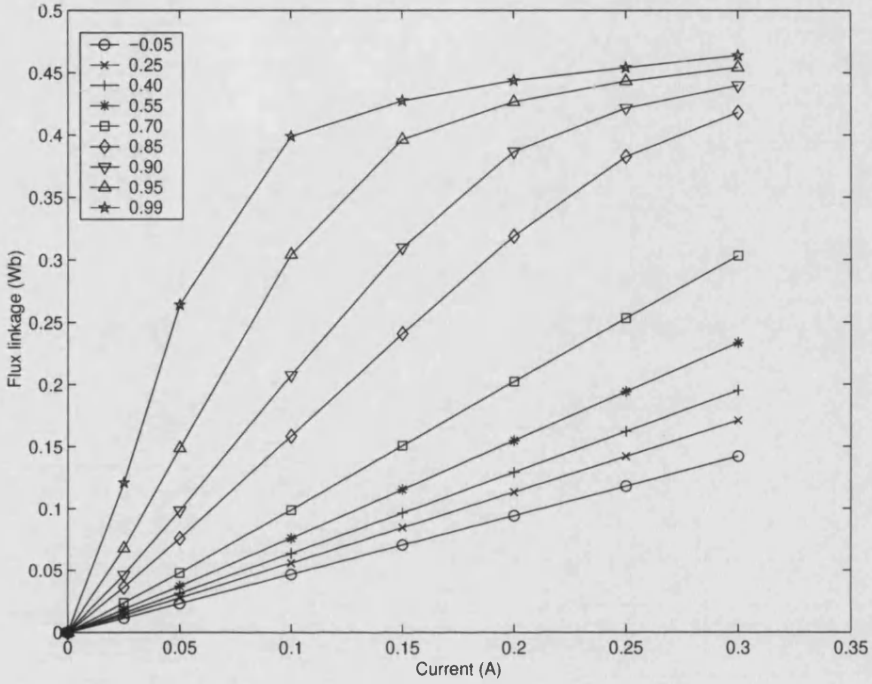


Figure 2.1: Flux linkage as a function of current for different armature positions (mm).

Table 2.1: Summary of table methods.

Calculated by FE	Transformed into
$\mathbf{A}(i,x), F_e(i,x)$	$\lambda(i,x), F_e(i,x)$
$W_m(i,x), W'_m(i,x)$	$\lambda(i,x), W'_m(i,x)$
$W'_m(i,x)$	$W'_m(i,x)$
$\mathbf{A}(i,x)$	$W_m(\lambda,x)$

Table 2.2: Position grid points, x(mm).

-0.05	0.25	0.40	0.55	0.70	0.85	0.90	0.95	0.99

Table 2.3: Current grid points, i(A).

0.0	0.025	0.05	0.1	0.15	0.2	0.25	0.3

2.3.6 Experimental verification

Tables

The modeling approaches discussed in the previous section were applied to a prototype linear actuator not very different in design to a solenoid valve, Fig.3.3. The characteristics of the actuator appear in [45]. To construct the required flux linkage, force, energy and coenergy tables, the position and current grid points shown in Tables 2.2 and 2.3 were used. The results of the FE calculations are presented in Tables 2.4 to 2.7. The FE package MEGA was used. Each one of the tables can be thought of as representing a surface; for instance, the flux linkage Table2.4 corresponds to the surface shown in Fig. 2.2. Fig. 2.1 is the plane graph equivalent to this surface.

Simulation

A closing transient was simulated and compared against measurements. The actuator is placed in an up-right position and the armature closes against its own weight. The stroke is 0.73mm. The position sensor exerts a small force in the direction of closing, and this was also introduced into the model [45]. The friction and damping

Table 2.4: Flux linkage $\lambda(i, x)$ in Wb, calculated from **A**. Rows have constant x.

0.0	.0115	.0230	.0464	.0701	.0939	.1178	.1418
0.0	.0137	.0275	.0556	.0842	.1129	.1418	.1707
0.0	.0155	.0312	.0633	.0960	.1289	.1620	.1950
0.0	.0184	.0371	.0757	.1149	.1545	.1941	.2334
0.0	.0236	.0479	.0986	.1503	.2021	.2531	.3031
0.0	.0363	.0757	.1582	.2406	.3188	.3827	.4180
0.0	.0463	.0984	.2073	.3095	.3866	.4215	.4398
0.0	.0679	.1487	.3042	.3963	.4265	.4431	.4542
0.0	.1208	.2640	.3988	.4273	.4436	.4542	.4639

Table 2.5: Electromagnetic force $F_e(i, x)$ in N. Rows have constant x.

0.0	.0692	.2776	1.122	2.559	4.596	7.236	10.48
0.0	.1265	.5085	2.070	4.746	8.539	13.46	19.52
0.0	.1876	.7560	3.101	7.127	12.85	20.28	29.41
0.0	.3091	1.251	5.202	11.99	21.67	34.23	49.52
0.0	.6105	2.507	10.62	24.69	44.69	70.10	100.5
0.0	1.796	7.826	34.17	79.12	138.9	200.4	239.7
0.0	3.182	14.40	64.10	142.8	223.0	265.6	289.5
0.0	7.573	36.36	152.2	58.90	298.9	322.3	338.5
0.0	26.45	126.0	285.8	326.8	350.9	366.5	380.9

Table 2.6: Magnetic energy $W_m(i, x)$ in mJ. Rows have constant x.

0.0	.1336	.5350	2.154	4.897	8.764	13.75	19.87
0.0	.1616	.6476	2.622	5.979	10.70	16.81	24.29
0.0	.1846	.7408	3.016	6.884	12.33	19.38	28.01
0.0	.2206	.8878	3.651	8.334	14.95	23.49	33.84
0.0	.2854	1.160	4.830	11.07	19.85	30.96	44.29
0.0	.4433	1.886	7.934	18.02	31.44	45.47	54.94
0.0	.5693	2.482	10.51	23.08	36.20	43.86	48.81
0.0	.8591	3.843	15.27	26.28	31.39	35.08	38.11
0.0	1.603	6.857	15.97	19.37	22.17	24.52	27.15

Table 2.7: Magnetic coenergy $W'_m(i,x)$ in mJ. Rows have constant x.

0.0	.1340	.5373	2.158	4.873	8.694	13.62	19.66
0.0	.1622	.6511	2.619	5.925	10.58	16.59	23.97
0.0	.1846	.7408	3.003	6.803	12.16	19.08	27.57
0.0	.2220	.8936	3.611	8.199	14.67	23.05	33.31
0.0	.2882	1.164	4.736	10.79	19.36	30.41	43.91
0.0	.4494	1.838	7.616	17.45	31.23	48.61	68.40
0.0	.5739	2.379	9.973	22.79	40.21	60.32	81.64
0.0	.8286	3.545	14.95	32.80	53.38	75.02	97.33
0.0	1.413	6.328	23.86	44.62	66.39	88.81	111.7

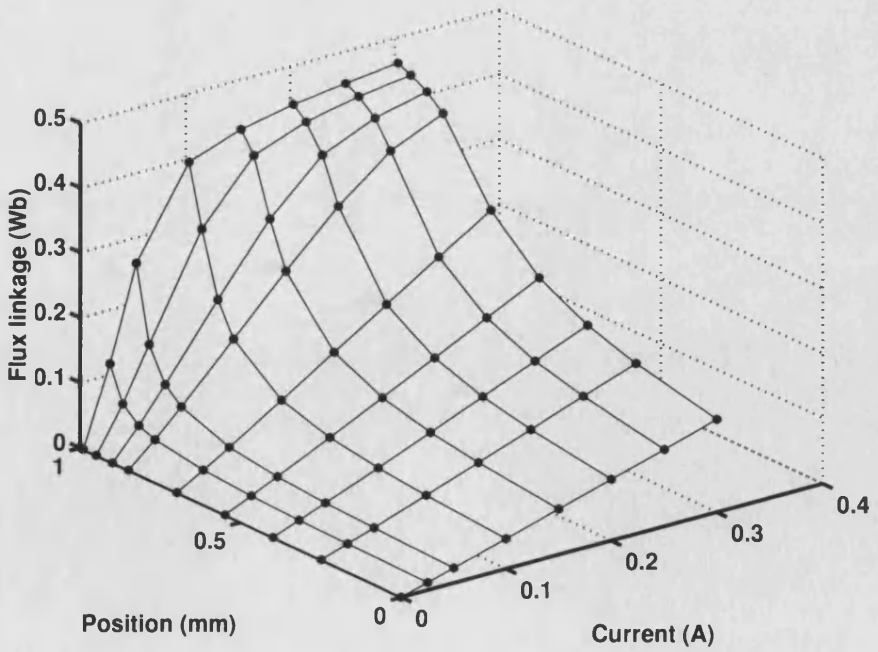


Figure 2.2: Flux linkage shown as a surface.

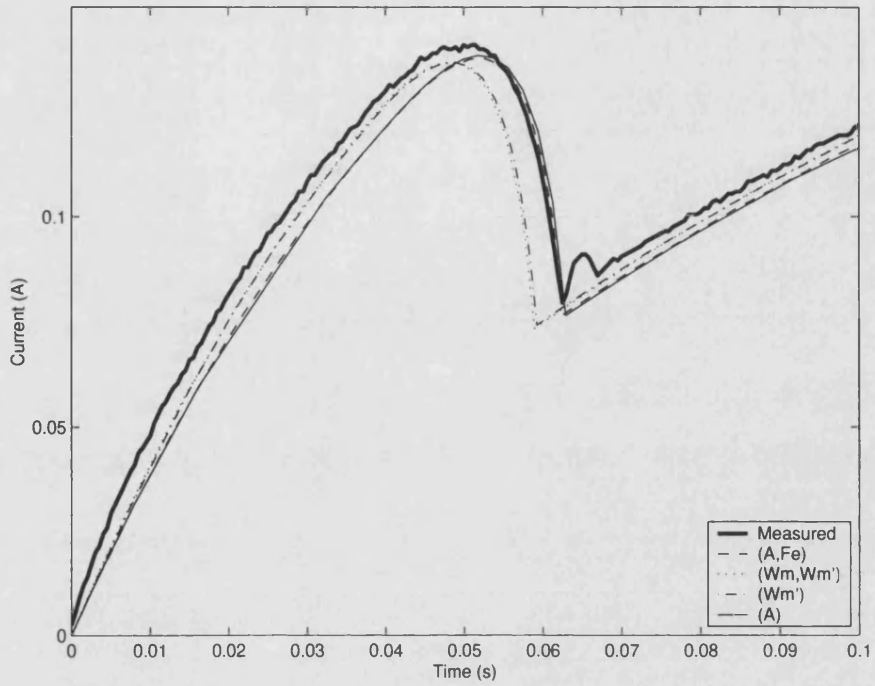


Figure 2.3: Transient current (A) vs. time (s) during closing.

were ignored. Fig.2.3 shows the measured and simulated currents, and the corresponding positions are shown in Fig.2.4. The results show a good correspondence. The elastic impact at the instant of closing has not been modeled, and this explains the difference between measured and simulated values around $t = 0.062$s.

The $(\mathbf{A}, F_e)$ and $(\mathbf{A})$ models have better accuracy than the (W_m, W'_m) and (W'_m) models. It can be expected that $(\mathbf{A}, F_e)$ will be more accurate since the force is evaluated directly from the given conditions; in contrast, the force is calculated as a numerical derivative for the (W_m, W'_m) and (W'_m) cases. However, the force for the $(\mathbf{A})$ case is also calculated by numerical differentiation and this seems to have no adverse effect.

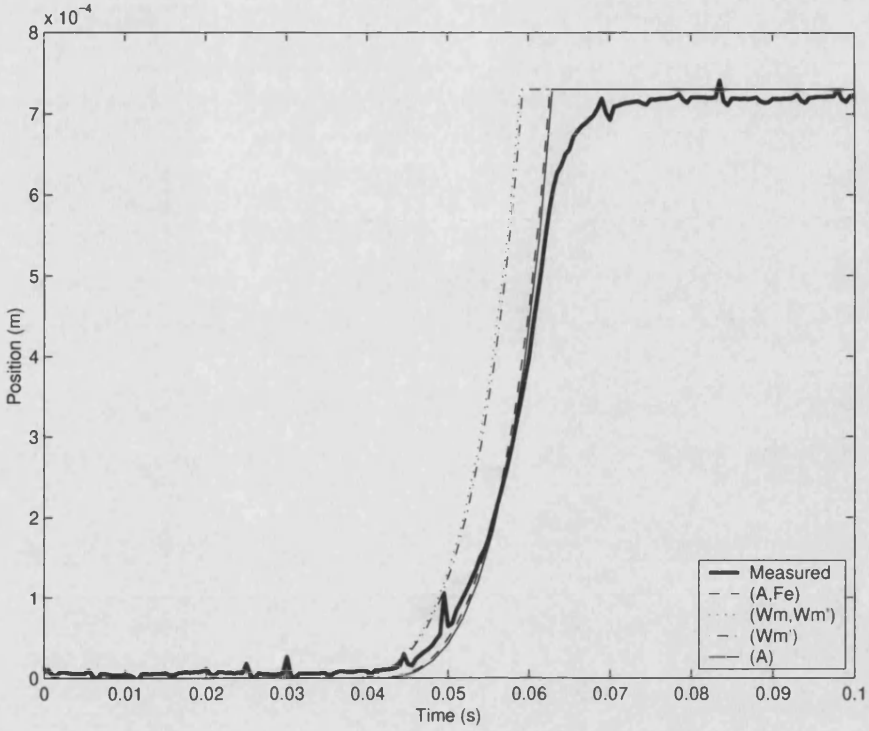


Figure 2.4: Transient position (m) vs. time (s) during closing.

2.3.7 Computational burden

It is interesting to evaluate the computational cost of the distinct methods. To that end, the closing transient was simulated using three time integration routines: a BDF (backward differentiation formula) code, an Adams-Moulton code and a Runge-Kutta code, all of them with error control. Partial derivatives are calculated by a finite difference formula in conjunction with spline interpolation.

The number of floating point operations required to complete the simulation was recorded, and the results are presented in Fig.2.5. It can be seen that the $(\mathbf{A})$ model requires the least number of operations for all three integration algorithms. The popular $(\mathbf{A}, F_e)$ method comes second, followed by (W_m, W'_m); the difference between these two is only the need to calculate the force by numerical differentiation

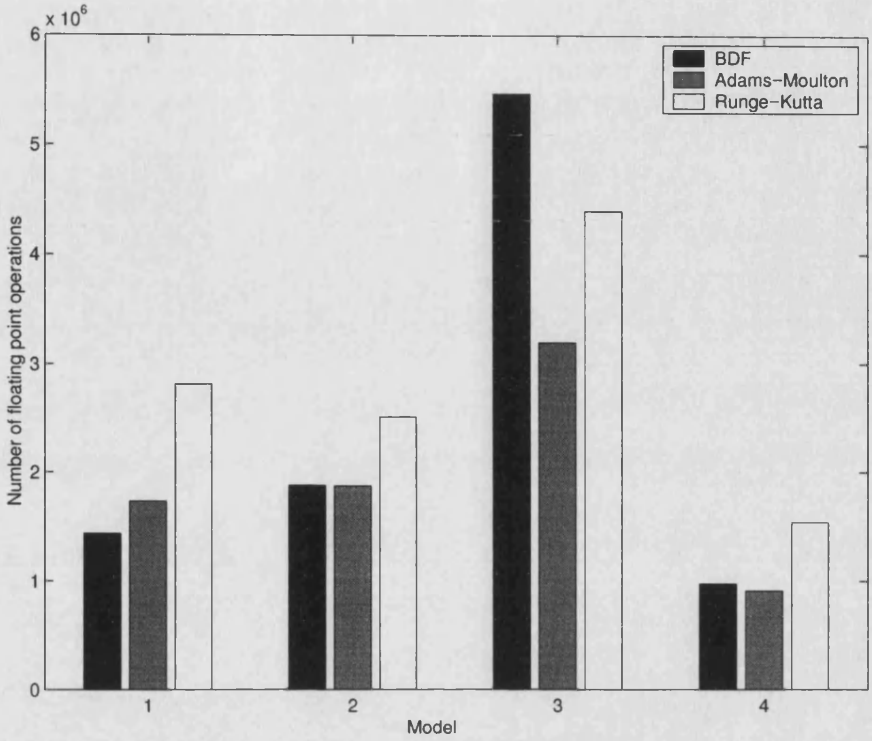


Figure 2.5: Number of floating point operations for the distinct models: 1-($\mathbf{A}, F_e$), 2-(W_m, W'_m), 3-(W'_m), 4-($\mathbf{A}$).

in the latter. (W'_m) is the method requiring the largest number of operations; this is because second order derivatives have to be calculated numerically so that the required number of interpolations is doubled.

A different transient condition, where the armature with damping moves against a spring force, was also simulated; the above results regarding the computational cost continue to hold. Thus it can be said that the ($\mathbf{A}$) model is the most efficient of the various alternatives considered. This is an unexpected feature of the method demanding the minimum resources from the FE program.

2.4 Table methods for n-coil devices

The energy procedure can be extended to the case of several electrical ports. The relevant relations for the (λ, x) formalism take the form:

$$dW_m = i_1 d\lambda_1 + i_2 d\lambda_2 + \ldots + i_n d\lambda_n - f dx, \tag{2.28}$$

$$i_j = \frac{\partial W_m}{\partial \lambda_j} \tag{2.29}$$

$$f = -\frac{\partial W_m}{\partial x} \tag{2.30}$$

For the (i, x) description, the equations are

$$dW'_m = \lambda_1 di_1 + \lambda_2 di_2 + \ldots + \lambda_n di_n + f dx, \tag{2.31}$$

$$\lambda_j = \frac{\partial W'_m}{\partial i_j} \tag{2.32}$$

$$f = \frac{\partial W'_m}{\partial x} \tag{2.33}$$

Table methods can be used in the more general case of a magnetic device with n coils. The complexity of the table and the number of static solutions required, however, become an important consideration. Practical experience indicates that table methods are still useful in the case of two-coil devices, but in the case of

three or more coils the book-keeping becomes too involved and the number of static solutions required grows dramatically.

$(\mathbf{A}, F)$ method

The $(\mathbf{A}, F)$ method can be applied to n-coil devices. The flux linkage for the first coil is given by

$$\lambda_1 = \lambda_1(i_1, i_2, ..., i_n, x) \tag{2.34}$$

and its circuit equation becomes

$$V_1 = R_1 i_1 + \frac{\partial \lambda_1}{\partial i_1}\frac{di_1}{dt} + \ldots + \frac{\partial \lambda_1}{\partial i_n}\frac{di_n}{dt} + \frac{\partial \lambda_1}{\partial x}\frac{dx}{dt}. \tag{2.35}$$

Similar equations exist for the remaining coils, so that the following matrix form can be written [57]:

$$\mathbf{V} = \mathbf{Ri} + \mathbf{L}\frac{d}{dt}\mathbf{i} + v\mathbf{s}, \tag{2.36}$$

where $\mathbf{L}$ is the inductance matrix given by

$$L_{pq} = \frac{\partial \lambda_p}{\partial i_q} \tag{2.37}$$

and $\mathbf{s}$ is the vector of velocity terms given by

$$s_p = \frac{\partial \lambda_p}{\partial x} \tag{2.38}$$

The state equation for the velocity and position is the same as for the one-coil case. The current derivatives have now a matrix coefficient, and this must be taken into account in the time stepping algorithm.

(W_m, W'_m) method

The (W_m, W'_m) formulation is only applicable to one-coil devices. In effect, (2.12) becomes

$$W_m + W'_m = \lambda_1 i_1 + \lambda_2 i_2 + \ldots + \lambda_n i_n, \tag{2.39}$$

so that the flux linkage for any coil cannot be uniquely determined from only this expression. Attempts to complement this equation lead to hybrid schemes (where additional equations are taken from one of the other methods discussed).

(W'_m) method

The (W'_m) method is directly applicable to the n-coil case. From (2.32) and (2.35), the circuit equation for the first coil is

$$V_1 = R_1 i_1 + \frac{\partial^2 W'_m}{\partial^2 i_1}\frac{di_1}{dt} + \ldots + \frac{\partial^2 W'_m}{\partial i_1 \partial i_n}\frac{di_n}{dt} + \frac{\partial^2 W'_m}{\partial i_1 \partial x}\frac{dx}{dt}. \tag{2.40}$$

Similar equations exist for the remaining coils, and all of them could be grouped in the matrix expression:

$$\mathbf{V} = \mathbf{R}\mathbf{i} + \mathbf{M}\frac{d}{dt}\mathbf{i} + v\mathbf{t}, \tag{2.41}$$

which is similar to (2.36) but now the matrix and vector entries are given by:

$$M_{pq} = \frac{\partial^2 W'_m}{\partial i_p \partial i_q}, \tag{2.42}$$

$$t_p = \frac{\partial^2 W'_m}{\partial i_p \partial x} \tag{2.43}$$

(A) method

The **(A)** method is still applicable. The transformation of the data from the FE solution is somewhat more involved, however. The key point is the function transformation (we consider a case with two coils to simplify the notation):

$$\begin{array}{ll} \lambda_1(i_1, i_2) & \\ & \rightarrow \quad i_1(\lambda_1, \lambda_2) \\ \lambda_2(i_1, i_2) & \\ & \qquad i_2(\lambda_1, \lambda_2). \end{array} \tag{2.44}$$

It is not immediately clear that this can be achieved, but performing the necessary calculations for a test case reveals that this is indeed possible. Figs. 2.6 and 2.7 show the contour plots for the flux linkages. It is clear that, for a given value of flux linkage, the curves intersect at a unique point, corresponding to a (i_1, i_2) pair, so that the function inversion is possible. Once the functions $i_1(\lambda_1, \lambda_2)$ and $i_2(\lambda_1, \lambda_2)$ have been obtained, the energy is calculated from [30]:

$$W_m(\Lambda_1, \Lambda_2, X) = \int_0^{\Lambda_1} i_1(\lambda_1, 0) d\lambda_1 + \int_0^{\Lambda_2} i_1(\Lambda_1, \lambda_2) d\lambda_2, \tag{2.45}$$

where capital letters indicate constant quantities.

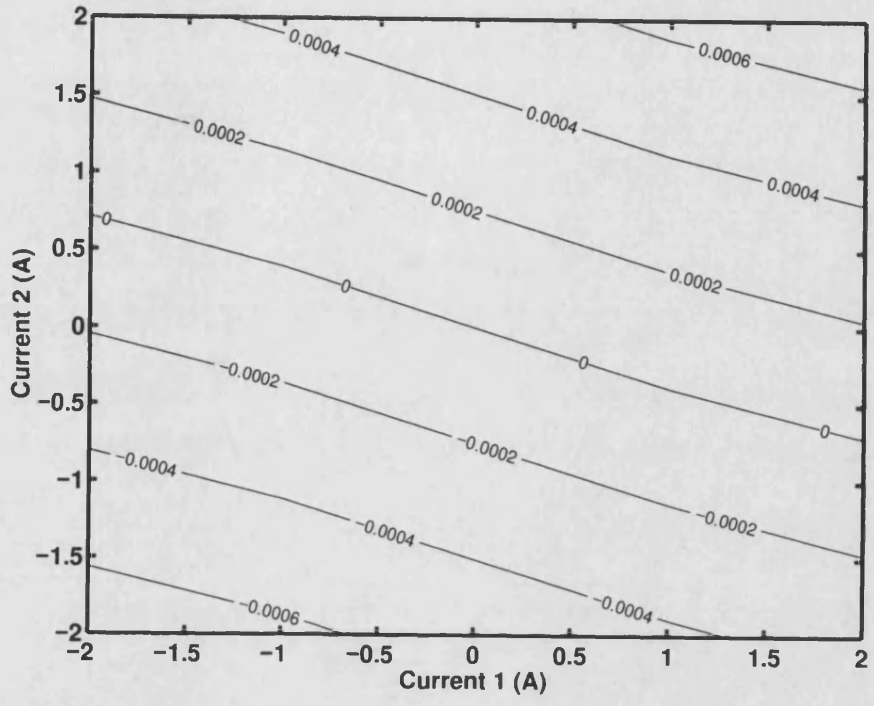


Figure 2.6: Contours of λ_1 for $x = 0$mm, device with two coils.

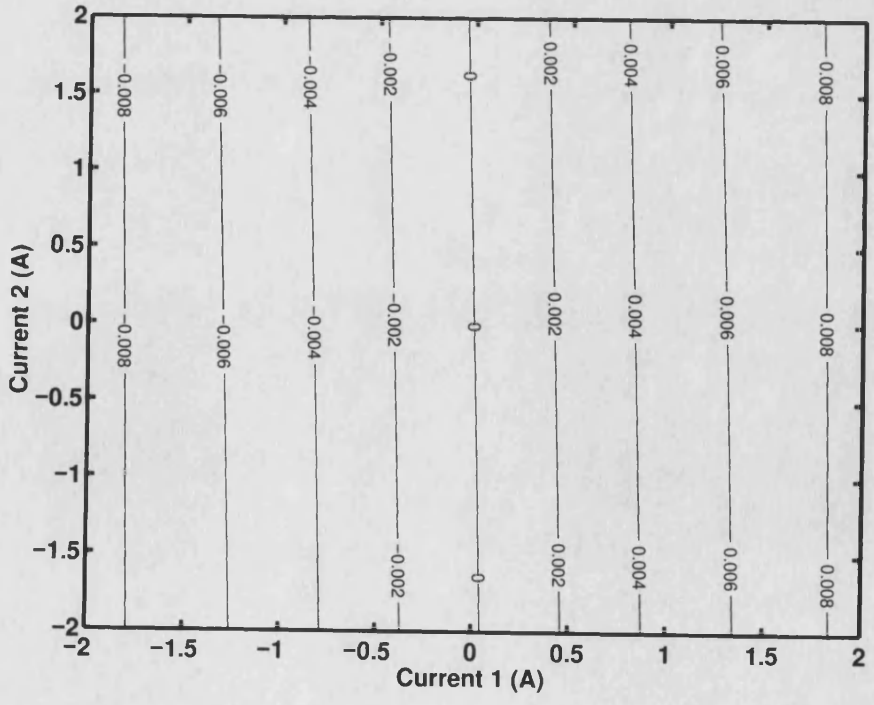


Figure 2.7: Contours of λ_2 for $x = 0$mm, device with two coils.

2.5 Conclusions

Energy methods provide a convenient method for reducing the complexity of finite element models while retaining the nonlinear behavior of the device. A table constructed by repeated finite element solutions covering the state space provides a complete description of the behavior of the device when used in conjunction with an interpolation scheme.

Two new table methods for single-coil devices were presented and compared against other existing approaches. The differences between them reside in the quantities selected as state variables and in the required amount of post-processing of the finite element solution. More importantly, the different methods demand different amounts of computing power in the actual simulation. The methods were compared in this respect by graphing the required number of arithmetical operations.

Comparisons with measured data for an experimental device were made and a good agreement was observed. The extension of the methods to the case of devices with multiple coils was discussed. The reported approaches have the advantage that physical meaning is retained by the reduced parameters, in contrast with purely algebraic reduction methods. The discussed system reduction is most useful in the case of negligible eddy currents, but consideration of the magnetic field diffusion can be made by adding a voltage-controlled resistance to the model.

Chapter 3

Rigid-body coupled problems

3.1 Introduction

The modeling of electromechanical problems involving rigid-body motion is discussed in this chapter. The simultaneous consideration of two distinct phenomena is required, as the evolution of the electromagnetic and the mechanical parts are influenced by each other. The finite element method is used to describe the electromagnetic part, and the rigid-body formulation of the equations of motion is used to describe the mechanical part. This approach can be thought of as a hybrid between a distributed-parameter model for the electromagnetic part and a lumped-parameter model for the mechanical part. The equations of the coupled problem are described and possible methods of solution are considered.

An underlying theme in the solution of the rigid-body coupled problem is the time-stepping scheme. We must guarantee that the solution process remains stable and the first requisite in this regard is to ensure that a stable time-stepping scheme is used. Only passing remarks on the subject will be made; a more complete discussion

can be found in a separate chapter.

This chapter is organized as follows. First, the origin of the system equations is discussed; then they are written in standard form to facilitate the discussion. Procedures for the solution of the coupled system are discussed; these include the Gauss–Seidel block iteration, the staggered solution process, and the full Newton iteration. Also presented is the theory behind matrix-free methods. Two new concepts are presented: the use of coupling variables to steer the coupled solution from a reduced equation set, and the interlacing of the nested solution processes to achieve a faster convergence.

Examples of coupled electromechanical problems are considered. First, static problems are tackled by using full Newton, block Gauss-Seidel and interlaced iterations. Dynamic problems are considered next; the question of the stability of the time-stepping scheme is decided on the grounds of the observed performance. Comparison between measured and calculated results are presented for a solenoid actuator and a rotational test rig. The long-term performance of some of the methods is investigated by applying them to the case of an induction motor.

3.2 System equations

Whatever the analysis method employed, be it a circuit (lumped parameter) model of the electromechanical device, or a field analysis method such as the finite element method, the resulting equations will be put in the following standard discrete form:

$$\mathbf{K}(\mathbf{x})\mathbf{x} + \mathbf{f}(\mathbf{x}) = \mathbf{0} \tag{3.1}$$

in the case of a static (equilibrium) problem, and

$$\mathbf{C}(\mathbf{x}, t)\dot{\mathbf{x}} + \mathbf{K}(\mathbf{x}, t)\mathbf{x} + \mathbf{f}(\mathbf{x}, t) = \mathbf{0} \tag{3.2}$$

in the case of a time transient (dynamic) problem. Moreover, the transient problem (3.2) can be transformed into the form (3.1) by using some suitable algorithm as discussed in a subsequent chapter. When we do so, we say that the equations have been time-discretized. Thus, we solve the time transient problem in a time-stepping sequence, where each step requires the solution of an equivalent (nonlinear) static problem.

3.2.1 Electromagnetic equations

The device's electromagnetic quantities are described in terms of fields by using the quasi-static magnetic subset of Maxwell's equations [30]:

$$\nabla \cdot \mathbf{B} = 0 \tag{3.3}$$

$$\nabla \times \mathbf{H} = \mathbf{J} \tag{3.4}$$

$$\nabla \cdot \mathbf{J} = 0 \tag{3.5}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{3.6}$$

In the case of moving conductors, the electric field is modified according to

$$\mathbf{E}' = \mathbf{E} + \mathbf{v} \times \mathbf{B} \tag{3.7}$$

Additionally, two macroscopic relations are used to model the material behavior:

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) \tag{3.8}$$

$$\mathbf{J} = \sigma \mathbf{E} \tag{3.9}$$

The process by which the field equations (3.3) to (3.6) are transformed into the discrete approximation (3.1) or (3.2) is discussed in general terms in [58]. More detailed treatments specially dealing with electromagnetics are found in [59, 60].

As non-conducting regions are likely to exist in the problem, the global damping matrix $\mathbf{C}$ will contain some rows of zeros (resulting from the region surrounding a node having zero conductivity) [61]. This imposes certain restrictions on the choice of a time stepping method.

3.2.2 Mechanical equations

The translational rigid-body equation of motion for a single mass is given by

$$M\ddot{x} + B\dot{x} + K(x - x_0) = F_e - F_g, \tag{3.10}$$

where x is the position, M is the mass of the body, B is the damping (friction) function, K is the spring constant, x_0 is the spring's neutral position, F_e is the force of electromagnetic origin and F_g is an external force function (for instance the weight of the body) [30]. All of these may be functions of the time t and/or of the position x or its derivative $\dot{x}$. A rotational system has a similar equation, with angular displacement θ instead of linear displacement x, moment of inertia J instead of mass M, etc.

The second-order equation (3.10) can be re-arranged in the required first-order form as follows:

$$\begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} B & K \\ -1 & 0 \end{bmatrix} \begin{bmatrix} v \\ x \end{bmatrix} + \begin{bmatrix} -Kx_0 - F_e + F_g \\ 0 \end{bmatrix} = \mathbf{0}, \tag{3.11}$$

where $v = \dot{x}$ is the velocity. A time-stepping algorithm will transform the first order equation (3.11) into a standard static-type equation. For instance, in the case of constant mechanical parameters, the backward Euler method yields

$$\begin{bmatrix} M + hB & hK \\ -h & 1 \end{bmatrix} \begin{bmatrix} v_{n+1} \\ x_{n+1} \end{bmatrix} + \begin{bmatrix} -Mv_n - hKx_0 - hF_e(t_{n+1}) + hF_g(t_{n+1}) \\ -x_n \end{bmatrix} = \mathbf{0}, \tag{3.12}$$

where h is the time step size. The determination of the next state $[v_{n+1}, x_{n+1}]$ starting from the known state $[v_n, x_n]$ requires the solution of the linear system (3.12).

3.2.3 Equations for the coupled system

In order to analyze the behavior of the electromechanical energy converter, both the electromagnetic equations and the mechanical equations must be solved simultaneously. Evaluation of the position requires the knowledge of the applied force of electromagnetic origin, but the evaluation of this force requires the solution of the field problem in the unknown position. For this reason, the problem is said to be coupled, meaning that two distinct physical domains determine the state of the system, and a separate solution in each domain is not possible.

After space and time discretization, the global equation can be written as

$$\begin{aligned} \mathbf{K}_e\mathbf{x}_e + \mathbf{f}_e &= \mathbf{0} \\ \mathbf{K}_m\mathbf{x}_m + \mathbf{f}_m &= \mathbf{0}, \end{aligned} \tag{3.13}$$

where the subindexes e and m stand for electromagnetic and mechanical system equations, respectively. This is a nonlinear problem in the vector variable $[\mathbf{x}_e, \mathbf{x}_m]$. Even if the decoupled problems are linear (an uncommon feature), the coupled problem becomes non-linear because of the interaction between domains. Thus, the general form of the vector nonlinear equation (3.13) is

$$\begin{aligned} \mathbf{F}_e(\mathbf{x}_e, \mathbf{x}_m) &= \mathbf{0} \\ \mathbf{F}_m(\mathbf{x}_e, \mathbf{x}_m) &= \mathbf{0}. \end{aligned} \tag{3.14}$$

3.3 Solution methods

In principle, any of the known methods of solution for nonlinear problems may be used to solve the coupled problem (3.14). However, the particular characteristics of coupled problems have led to the search for specialized algorithms. Some of the possibilities are considered next. Most of the discussion may be applied to other types of coupling.

3.3.1 Simple iteration

The time-discretization scheme used to go from (3.2) to (3.1) introduces its own stability constraints in the solution process. If the nonlinear coupled equations (3.14) are not satisfied at each time step, the stability will be compromised. Thus, it is desirable to arrive to a consistent solution in order to confine any instability to a known source (the time stepping algorithm) so that, hopefully, if the time-stepping method is stable, so will be the coupled solution.

A very direct approach to arrive to a consistent solution of (3.14) solves alternately the electromagnetic equation and the mechanical equation , since a linear equation solver can be applied directly to these equations without further modification. This is the equivalent of the fixed-point iteration for single domain problems and relies in the existence of single-domain solvers of the form:

$$\mathbf{x}_e^{k+1} = \mathbf{S}_e(\mathbf{x}_e^k, \mathbf{x}_m) \tag{3.15}$$

$$\mathbf{x}_m^{k+1} = \mathbf{S}_m(\mathbf{x}_e, \mathbf{x}_m^k) \tag{3.16}$$

Two variants are recognized: the Jacobi and Gauss–Seidel block iterations. The Jacobi iteration is:

$$\begin{aligned} \mathbf{x}_e^{k+1} &= \mathbf{S}_e(\mathbf{x}_e^k, \mathbf{x}_m^k) \\ \mathbf{x}_m^{k+1} &= \mathbf{S}_m(\mathbf{x}_e^k, \mathbf{x}_m^k), \end{aligned} \tag{3.17}$$

and is useful in the case of parallel computer architectures, since the states can be updated independently. The Gauss–Seidel iteration is:

$$\begin{aligned} \mathbf{x}_e^{k+1} &= \mathbf{S}_e(\mathbf{x}_e^k, \mathbf{x}_m^k) \\ \mathbf{x}_m^{k+1} &= \mathbf{S}_m(\mathbf{x}_e^{k+1}, \mathbf{x}_m^k), \end{aligned} \tag{3.18}$$

and is expected to converge faster.

The Jacobi and Gauss–Seidel iterations have some drawbacks: they may fail to converge, and even if they do, the number of iterations required can be quite large (as compared to other methods).

3.3.2 Staggered solution

One frequently used strategy consists in solving the electromagnetic equation for a given position $\mathbf{x}_m^k$, from which $\mathbf{x}_e^k$ and the force of electromagnetic origin F_e^k are obtained. Then, considering this force as a constant over the time step, the next position $\mathbf{x}_m^{k+1}$ is calculated by solving the mechanical equation [62]. This is essentially a one-step block Gauss–Seidel scheme, and most probably the approximation $[\mathbf{x}_e^{k+1}, \mathbf{x}_m^{k+1}]$ is *not* a solution of the coupled nonlinear equation (3.14). As a result accuracy problems may appear and a small time step size is often required to avoid them. The staggered scheme is said to provide a 'weak' coupling.

3.3.3 Newton iteration

It is well known that a faster convergence rate is achieved by using the tangent or Jacobian matrix. The Newton iteration solves repeatedly the system

$$\begin{bmatrix} \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_e} & \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_m} \\ \frac{\partial \mathbf{F}_m}{\partial \mathbf{x}_e} & \frac{\partial \mathbf{F}_m}{\partial \mathbf{x}_m} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_e \\ \Delta \mathbf{x}_m \end{bmatrix} = \begin{bmatrix} -\mathbf{F}_e \\ -\mathbf{F}_m \end{bmatrix}. \tag{3.19}$$

The size of the linear equation now is the sum of the sizes of the domain problems (the size of the problem is increased). Also, we now need to calculate the Jacobian, which is problem dependent and thus cannot be applied to other kinds of coupling. The Newton scheme is said to provide a 'strong' coupling.

3.3.4 Coupling variables

For certain classes of problems, the interaction between physical domains takes place not between the primary or domain variables, but by means of a function of them. For instance, in the coupling of a magnetic system with a rigid body system, it is clear that the individual values of the magnetic vector potential $\mathbf{A}$ are not directly relevant to the mechanical response, but the resultant force is. This force is given by an integral function of $\mathbf{A}$.

Suppose we attempt to solve this kind of problem by a Gauss–Seidel iteration. The sequence could be, for each time step:

1. Set the initial mechanical state $\mathbf{x}_m$.

2. Solve the magnetic problem to get the magnetic vector potentials and from

them the total force F_e.

3. Solve the mechanical equation, regarding the force as a constant.

4. Repeat until convergence is achieved.

This iterative scheme may be slow or fail to converge. The important point to observe is that there is a definite set of variables communicating between 'modules': the mechanical variables are the input to the field model and the result is a scalar force. This force is the input of the mechanical model, and the output is a set of corrected mechanical variables. Thus, the coupled problem can be stated in terms of these 'interface' or 'coupling' variables. What we seek is a set $\mathbf{x}$ and F satisfying the nonlinear equation

$$\begin{aligned} F - F_e(\mathbf{x}) &= 0 \\ \mathbf{x} - \mathbf{x}_m(F) &= \mathbf{0}. \end{aligned} \tag{3.20}$$

In these equations, $F_e(\mathbf{x})$ is a transformation rule $\mathbf{x} \rightarrow F$ and the subscript e is used to denote that this transformation is achieved by solving the electromagnetic field equations. Similarly, $\mathbf{x}_m(F)$ performs the transformation $F \rightarrow \mathbf{x}$, m indicating that this transformation is achieved by solving the mechanical equation.

A Newton iteration can be used instead of the Gauss-Seidel scheme [63]. In such case we have:

$$\begin{bmatrix} 1 & -\frac{\partial F_e}{\partial \mathbf{x}} \\ -\frac{\partial \mathbf{x}_m}{\partial F} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta F \\ \Delta \mathbf{x} \end{bmatrix} = \begin{bmatrix} -F + F_e(\mathbf{x}) \\ -\mathbf{x} + \mathbf{x}_m(F) \end{bmatrix}. \tag{3.21}$$

It is worth noting that the details of the scheme are the same for other cases of coupling between field and lumped parameter models. For example, coupled circuit–

field problems.

3.3.5 Newton variants

Some variants of the Newton iteration have been proposed in order to overcome its disadvantages; for instance, the Jacobian matrix may be dense or be difficult to calculate explicitly. Also, individual solvers may exist for each domain and it would be a good idea to use them without change.

The quasi-Newton methods attempt to achieve a fast convergence by approximating the Jacobian instead of calculating it [64]. It appears that this approach has not been attempted for coupled problems, neither by using a global approximation to the Jacobian nor by the hybrid method of calculating exactly the diagonal blocks and approximating the non-diagonals.

A related method avoids the calculation of the Jacobian and its storage by estimating the matrix-vector products required by a Krylov-subspace linear equation solver (CG, GMRES, etc.). The technique is called matrix-free and has been used in several instances of coupled fields [65, 66, 67, 63]. A disadvantage is that it may be difficult to select an appropriate value of the matrix-free parameter. Also, care must be exercised in the choice of preconditioner. A justification of this approach is presented now, based on the concept of directional derivative. The motivation for the matrix-free approach is the need to solve systems of nonlinear equations in single and coupled-field problems. A Jacobian matrix must be calculated and assembled, and this may represent an expensive task; moreover, it requires special-purpose code which can be difficult or slow to implement, specially for coupled problems. On the other hand, iterative techniques for the solution of the resultant linear equation

require the evaluation of Jacobian–vector products. This can be used to reduce the computational burden in the following way: the Jacobian matrix is not assembled, since we are only interested in the Jacobian-vector product, which is approximated by a finite difference expression. The (algebraic) nonlinear problem to be solved may be written in the general form

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0. \end{aligned} \tag{3.22}$$

The k-th row of the Jacobian matrix is

$$\left[\frac{\partial f_k}{\partial x_1}\frac{\partial f_k}{\partial x_2}\cdots\frac{\partial f_k}{\partial x_n}\right]. \tag{3.23}$$

The product of the k-th row of the Jacobian matrix by the vector $\mathbf{s} = [s_1 s_2 \ldots s_n]$ may be interpreted as a directional derivative, that is, a gradient dotted with the displacement, or as a first order approximation of the increment df_k:

$$df_k = \frac{\partial f_k}{\partial x_1}s_1 + \frac{\partial f_k}{\partial x_2}s_2 + \ldots + \frac{\partial f_k}{\partial x_n}s_n. \tag{3.24}$$

This increment df_k is evaluated by using

$$df_k = f_k(x_1 + s_1, x_2 + s_2, \ldots, x_n + s_n) - f(x_1, x_2, \ldots, x_n), \tag{3.25}$$

but since f_k is a nonlinear function, the approximation (3.24) may not be adequate. More critically, it may not truly approach (3.25), especially for large displacements s_i. In order for the directional derivative to be evaluated more accurately, a small

displacement is needed and we could write

$$df_k \approx f_k(x_1 + \epsilon s_1, x_2 + \epsilon s_2, \ldots, x_n + \epsilon s_n) - f_k(x_1, x_2, \ldots, x_n), \tag{3.26}$$

where the parameter ϵ is a very small number. This expression approximates

$$df_k = \frac{\partial f_k}{\partial x_1}\epsilon s_1 + \frac{\partial f_k}{\partial x_2}\epsilon s_2 + \ldots + \frac{\partial f_k}{\partial x_n}\epsilon s_n. \tag{3.27}$$

The required quantity (3.24) can thus be approximated by

$$df_k \approx \frac{1}{\epsilon}\left[f_k(x_1 + \epsilon s_1, x_2 + \epsilon s_2, \ldots, x_n + \epsilon s_n) - f_k(x_1, x_2, \ldots, x_n)\right]. \tag{3.28}$$

The product $\mathbf{Js}$ used in the Newton-Raphson iteration can then be estimated as

$$\mathbf{Js} = \frac{\mathbf{f}(\mathbf{x} + \epsilon\mathbf{s}) - \mathbf{f}(\mathbf{x})}{\epsilon} \tag{3.29}$$

3.3.6 Interlaced iteration

The implementation of the Jacobi or Gauss–Seidel processes is straightforward if separate solvers (3.15), (3.16) exist for each domain, and these solvers may be treated as black boxes. Internally, each solver may use itself a fixed point, Newton or quasi-Newton iteration. The whole process may thus be seen as one of nested iterations. From the viewpoint of the global iteration, the choice of algorithm for the internal solvers is not of interest. However, since each global iteration requires several internal iterations, a fast convergence is desirable.

Another possibility arises if only one internal iteration of the decoupled problems is

calculated alternately. We will call this an interlaced iteration. The idea appears in [68, 69]. The process can be seen as a variant of the block simple iteration methods discussed above. In effect, it shares the same general procedure, but updates the states even when the domain variables have not converged. It also shares some characteristics of the quasi-Newton methods because it is equivalent to the use of the Jacobian approximation

$$\mathbf{J} \approx \begin{bmatrix} \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_e} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{F}_m}{\partial \mathbf{x}_m} \end{bmatrix}. \tag{3.30}$$

Consider the smooth surface $f(x_e, x_m)$ shown in Fig.3.1. The solution of (3.19) provides the direction and longitude of a displacement $[\Delta x_e, \Delta x_m]$ such that the approximation to $f(x_e, x_m) = 0$ is improved. In graphical terms, this represents a transversal displacement from P_1 to P_2, where we expect $f(P_2)$ to be closer to the solution.

The graphical equivalent of the interlaced iteration is a displacement along the direction x_e, followed by a displacement in the direction x_m, as illustrated in Fig.3.2. Each of these displacements improves the approximation, and the combined effect is hopefully not very different to the transversal displacement given by the complete Jacobian.

We can as well consider the Taylor series expansion of $f(x_e, x_m)$ (on which the Newton iteration is based) to first order:

$$f(x_e + \Delta x_e, x_m + \Delta x_m) = f(x_e, x_m) + \frac{\partial f}{\partial x_e}\Delta x_e + \frac{\partial f}{\partial x_m}\Delta x_m \tag{3.31}$$

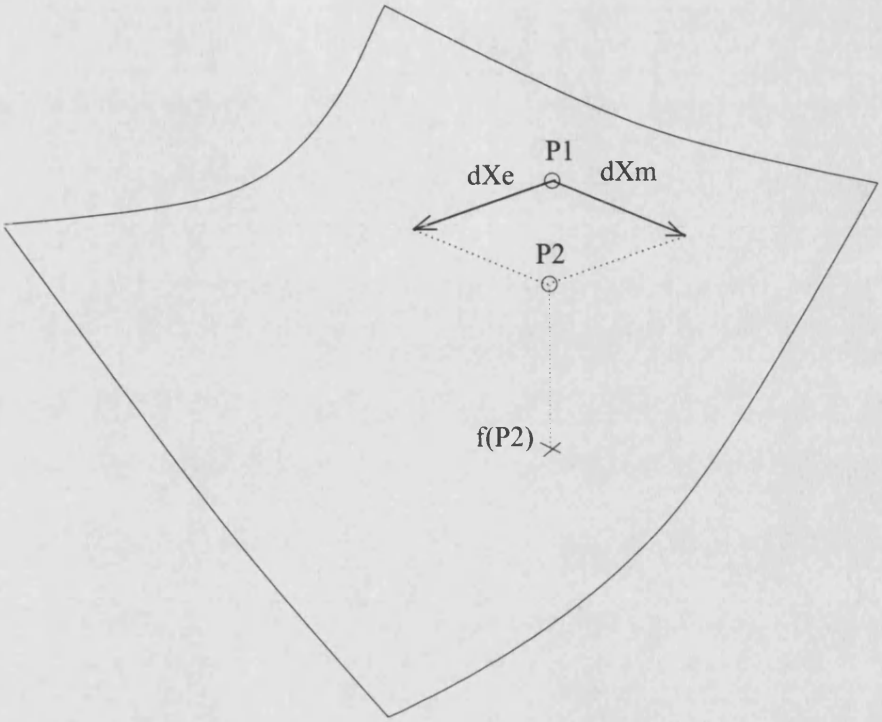


Figure 3.1: Graphical interpretation of Newton iteration.

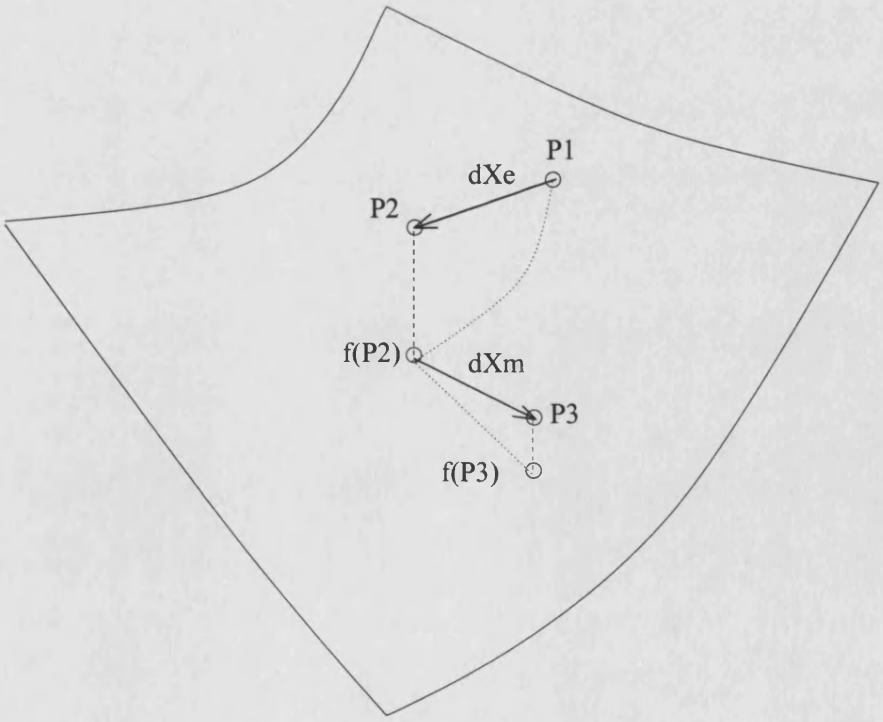


Figure 3.2: Graphical interpretation of interlaced iteration

If we decide to travel along x_e only, then $\Delta x_m = 0$ and

$$f(x_e + \Delta x_e, x_m) = f(x_e, x_m) + \frac{\partial f}{\partial x_e}\Delta x_e, \tag{3.32}$$

which provides an equation for Δx_e when equated to zero. The expression (3.32) is an approximation to $f(x_e, x_m)$ of the same order (first) as (3.31), and as long as we do not change x_m, no further error is introduced. For sufficiently small Δx_e if any improvement in the solution is achieved by traveling along x_e, it will be given by (3.32). We may add that if the surface is sufficiently smooth for the Newton iteration to converge, we expect the interlaced iteration to converge too.

3.4 Static problems

3.4.1 An equilibrium problem

We apply the ideas of the previous sections to the problem of a solenoid magnet excited by a constant current, trying to pull an armature against the opposing force of a spring. For the calculation, we use the solenoid actuator outlined in Fig.3.3 and described in [45]. The actuator, being of the attraction type, has a force characteristic which varies inversely with the air gap length; meanwhile the spring force function is a straight line. Using the definitions of Fig. 3.4, the situation can be depicted graphically as shown in Fig. 3.5 where the magnet force F_e and the negative of the spring force, $-F_m$, have been plotted. A position of equilibrium will be reached when the magnet force equals the spring force:

$$F_e(x) = F_m(x), \tag{3.33}$$

which can be either at x_a or x_b in Fig.3.5. Lets examine the neighbourhood of these points. In a position for which $x < x_a$, the magnet force is larger than the spring force, and as a result the distance x will increase (this has been indicated by an arrow in the figure). The opposite will happen for a point such that $x_a < x < x_b$: in this case the spring force is larger than the magnet force so that the distance x of the armature will decrease. Thus, x_a is a stable equilibrium point. Proceeding similarly, it may be seen that x_b is an unstable equilibrium point, since a small variation which makes x decrease results in the armature taking the stable position x_a, while an increase to x from x_b results in the armature 'escaping' and smashing against the magnet pole.

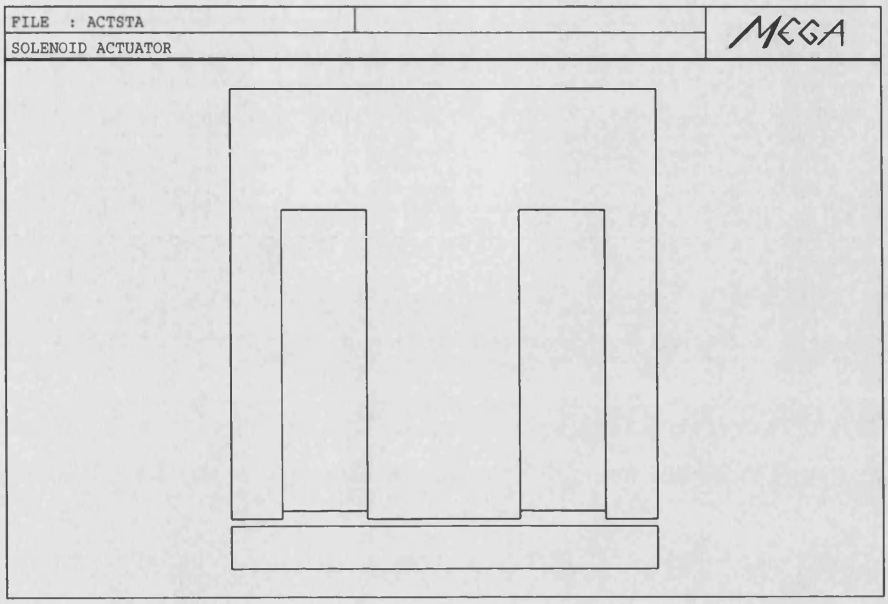


Figure 3.3: Outline of solenoid actuator.

Let us specify an initial position $x = 0$, and solve the coupled problem to obtain the stable position x_a.

3.4.2 Solution with strong coupling

We use the strong coupling approach (Newton iteration) and the concept of coupling variables of Section 3.3.4. In this case, the set of mechanical variables $\mathbf{x}_m$ reduces to the scalar position x. The iteration equation is then

$$\begin{bmatrix} 1 & -\frac{\partial F_e}{\partial x} \\ -\frac{\partial x_m}{\partial F} & 1 \end{bmatrix} \begin{bmatrix} \Delta F \\ \Delta x \end{bmatrix} = \begin{bmatrix} -F + F_e(x) \\ -x + x_m(F) \end{bmatrix}. \tag{3.34}$$

The Jacobian terms must now be calculated. For a linear spring, the force-distance relationship has the form

$$x_m = KF + x_0 \tag{3.35}$$

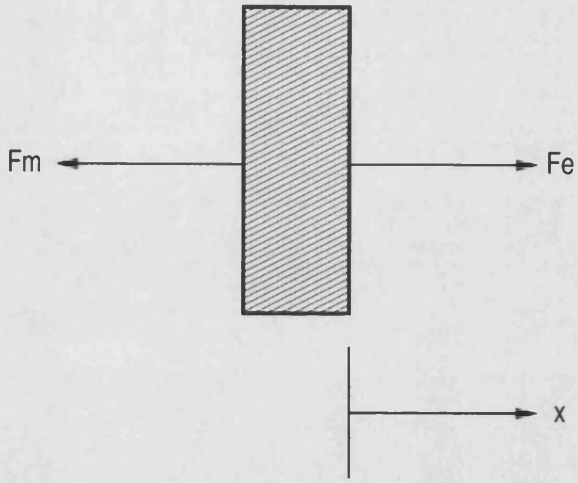


Figure 3.4: Forces on armature: F_e is the magnet force, F_m is the spring force. Distance x is measured from the spring neutral position

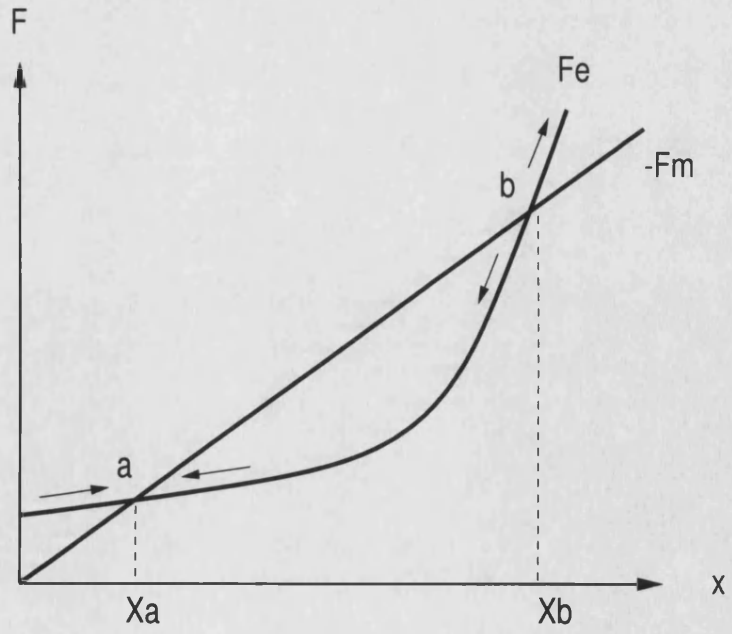


Figure 3.5: Stability characteristics of equilibrium problem.

from where the corresponding Jacobian term is

$$\frac{\partial x_m}{\partial F} = K. \tag{3.36}$$

The other Jacobian term, the stiffness $\frac{\partial F_e}{\partial x}$, is more difficult to calculate since the solution of the field problem is required. A finite difference approximation can be used, according to

$$\frac{\partial F_e}{\partial x} = \frac{F_e(x + \Delta x) - F_e(x)}{\Delta x}. \tag{3.37}$$

The question of the magnitude of the step Δx must be solved. The values of force for several step magnitudes at $x = 0$ are shown in Table 3.1, and we select $\Delta x = 1\mu$m. The Newton iteration proceeds as shown in Table 3.2. We see that a few Newton iterations are sufficient to solve the problem; however, each iteration requires two complete solutions of the nonlinear magnetic domain because the finite difference approximation of the stiffness requires two points. In turn each solution of the magnetic problem requires several internal Newton iterations and it may be thought that this is an expensive approach. We will see next how to reduce the computational cost.

$\Delta x(\mu\text{m})$	F_e(N)	$\frac{\partial F_e}{\partial x}$ (N/μm)
0	13.1312	—
1	13.1636	0.032400
2	13.1964	0.032460
3	13.2286	0.032467

Table 3.1: Finite difference approximation of stiffness.

Iteration (k)	F_e^k(N)	x^k (μm)	ΔF_e(N)	Δx (μm)
1	0.0	0.0	29.5976	508.221
2	29.5976	508.221	84.9632	23.6009
3	114.561	531.822	3.21293	1.43948
4	117.774	533.262	-0.63237	-0.72203
5	117.141	532.540	—	—

Table 3.2: Results of Newton iteration for position problem.

3.4.3 Solution with simple and interlaced iteration

Tables 3.3 and 3.4 show the cumulative number of linear equation solutions for the magnetic problem when the Gauss–Seidel and the interlaced iterations are used, respectively. It turns out that the both of them solve the problem very efficiently, in a number of steps similar to the Newton method of the preceding Section. The computational cost, however is smaller.

The Gauss–Seidel and interlaced approaches solve the problem in an equal number of iterations, but the number of linear equation solutions required in each case differs greatly. The reason is that the interlaced iteration updates the position after calculating only one Newton iteration of the nonlinear magnetic problem, while in the Gauss–Seidel iteration full convergence of this internal iteration is waited. As can be appreciated in Fig. 3.6, the force is very well estimated even at the first internal iteration. The number of linear equation solutions is compared in Fig. 3.7. The number of linear equation solutions for the Newton process of the preceding section is twice that of the Gauss–Seidel scheme.

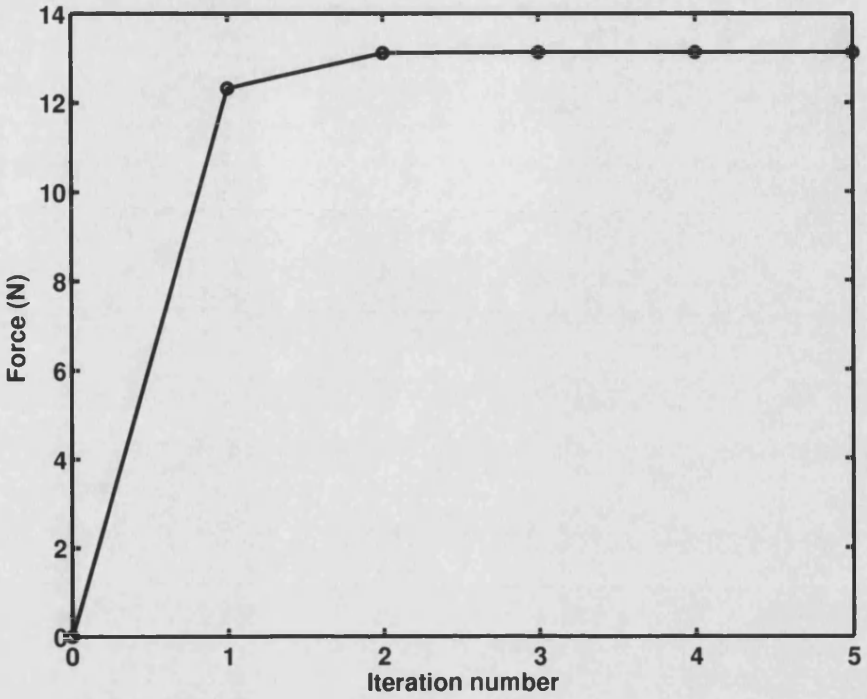


Figure 3.6: Force for the first internal nonlinear solution.

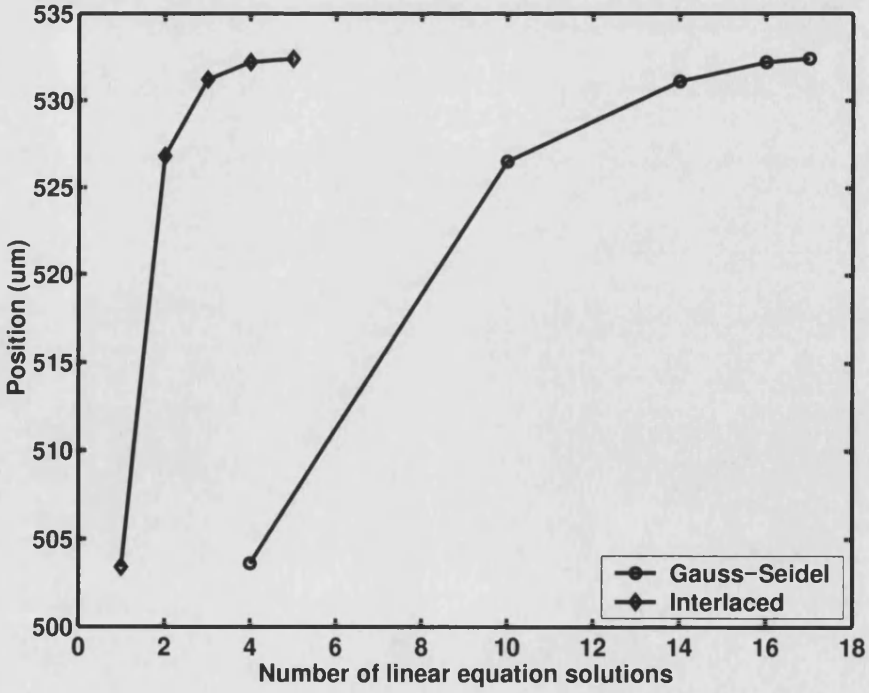


Figure 3.7: Number of linear equation solutions.

Iteration k	F_e (N)	x (μm)	Lin.Eq.Solutions (cumulative)
1	13.1	503.6	4
2	95.5	526.5	10
3	112.1	531.1	14
4	115.9	532.2	16
5	116.8	532.4	17

Table 3.3: Results of Gauss–Seidel iteration for position problem.

Iteration k	F_e (N)	x (μm)	Lin.Eq.Solutions (cumulative)
1	12.30	503.4	1
2	96.50	526.8	2
3	112.5	531.2	3
4	116.0	532.2	4
5	116.8	532.4	5

Table 3.4: Results of interlaced iteration for position problem.

3.4.4 An excitation problem

With the same actuator studied in the preceding sections, an interesting setup arises if we ask the current required to produce a given force at certain position. The equation to solve now reduces to

$$F - F_e(i) = 0. \tag{3.38}$$

Since there is now only one scalar equation, it is not possible to apply the Gauss–Seidel method or the interlaced method and the problem must be solved by a Newton iteration. Of course, a graph of the force as a function of the current may be constructed and the solution approximated from it, but a numerical solution is preferred. The problem illustrates also a curious mapping of a scalar quantity to a vector and then again to a scalar:

$$i \to \mathbf{A} \to F_e \tag{3.39}$$

The Newton iteration equation is

$$-\frac{\partial F_e}{\partial i}\Delta i = -F + F_e(i). \tag{3.40}$$

The derivative in the last equation is approximated by finite differences. For a desired force $F_e = 175$N, selecting an initial current $i = 0$ leads us immediately into trouble: the Newton iteration does not converge, as can be appreciated in Table 3.5. The graph of the force as a function of the current in Fig. 3.8 provides an explanation: for the selected initial current, the derivative is very small and as a result the corrected current will be very large, falling in a region where the derivative is again small. As a result, the subsequent current values oscillate between positive and negative values, trapped in the upper arms of the curve.

A more sensible choice of initial current is $i = 0.1$A. In this case, the process converges in a few iterations, shown in Table 3.6.

Iteration (k)	i^k(A)	Δi(A)
1	0.0	9.542
2	9.542	-16.457
3	-6.915	14.119
4	7.203	-14.409
5	-7.206	14.341

Table 3.5: Results of Newton iteration for current problem, initial current $i = 0$(A).

Iteration (k)	i^k(A)	Δi(A)
1	0.1	0.325
2	0.425	-0.245
3	0.180	0.125
4	0.305	-0.019
5	0.286	0.002
6	0.288	-0.0006

Table 3.6: Results of Newton iteration for current problem, initial current $i = 0.1$(A).

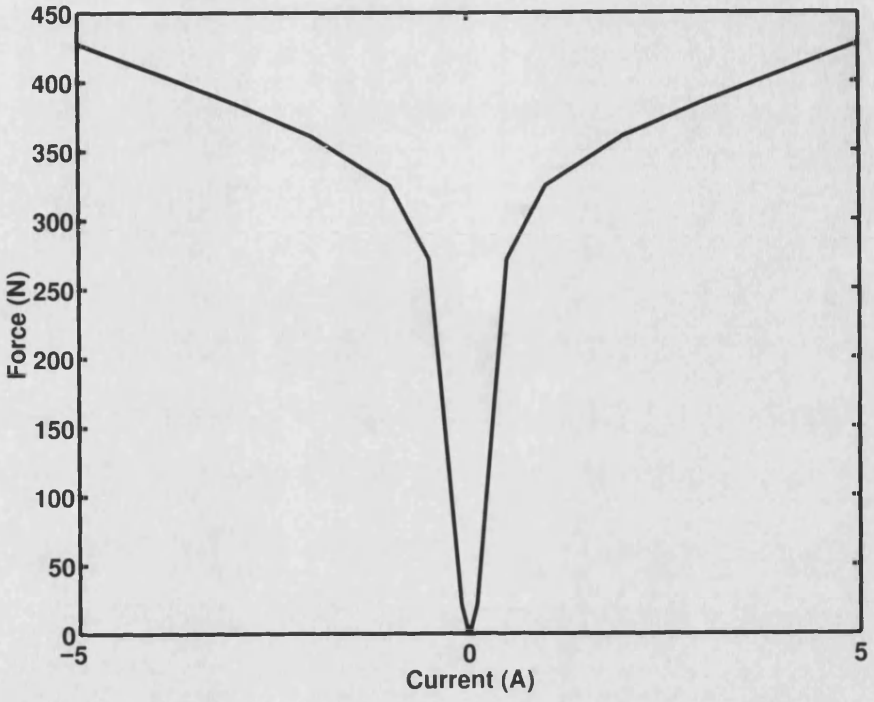


Figure 3.8: Actuator force characteristic.

3.5 Dynamic problems

3.5.1 Solenoid actuator

The ideas discussed in the previous sections will be used to solve a dynamic coupled problem involving the solenoid actuator of Fig.3.3. The moving structure is modeled as a rigid body, according to (3.10). The mechanical system parameters were chosen so that there is a certain amount of damping and the spring force is high. Once the coil is energized by a step voltage source, the armature moves from the reference position to the final position without actually closing. The electromagnetic part was modeled by the finite element method; the movement is taken into account by deforming the air gap elements; this is possible since the displacements are always small enough to avoid severe distortion.

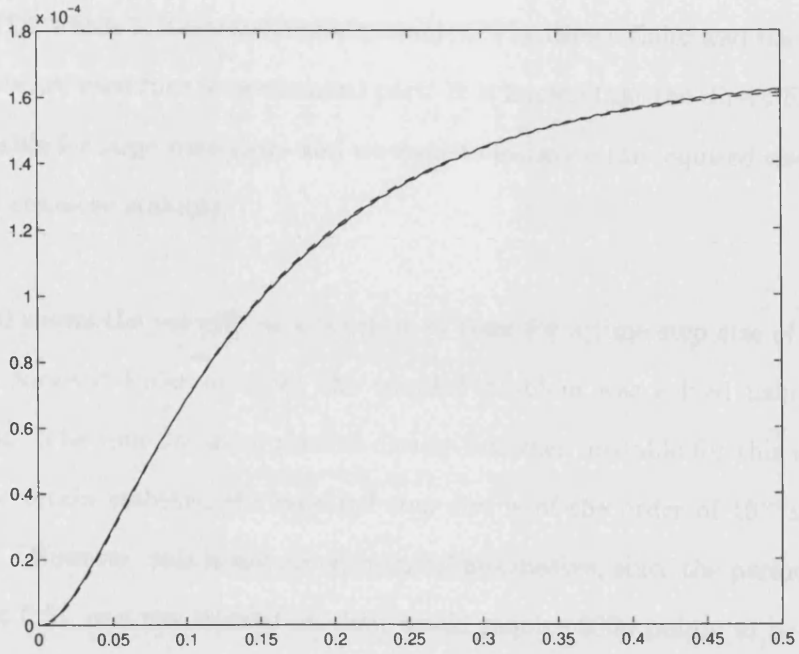


Figure 3.9: Position (m) vs. time (s): (–) Lumped-parameter model, (- -) Finite element coupled model with $\Delta t = 10^{-3}$s.

A lumped-parameter (table look-up) model for the same device is used as reference to evaluate the results; this is possible since the eddy currents do not play an appreciable role in this particular case. The time-integration scheme used for the lumped-parameter model uses a variable time step size, ensuring a low error. The results from the backward Euler method with $\Delta t = 10^{-3}$s are practically the same as those for the lumped-parameter model, as seen in Fig.3.9. Results from both the lumped-parameter and the finite element models compare well against measurements taken on a prototype actuator, as shown in the next section.

Stability

The first question to be addressed is that of the stability of the numerical method. The θ-method is used as time-stepping algorithm for the electromagnetic part, with

$\theta = 1$ (for which it is unconditionally stable). The direct Euler and backward Euler methods are used for the mechanical part. It is known that the direct Euler method is unstable for large time steps and we want to estimate the required size of the time step to conserve stability.

Fig.3.10 shows the velocity as a function of time for a time step size of $\Delta t = 10^{-3}$s for the forward Euler method; the coupled problem was solved using interlaced iteration. The time-stepping process clearly becomes unstable for this step size. In order to retain stability, the required step size is of the order of 10^{-4}s, as seen in Fig.3.11. However, this is not an economical alternative, since the period of interest is about 0.5s, and the simulation then would require 5000 points to be calculated, even if the variables follow very smooth curves.

The backward Euler method offers better stability characteristics. As seen in Figs.3.12 to 3.14, the algorithm remains stable even with a very large time step. On the other hand, the accuracy of the solution is seen to depend on the step size, a characteristic of all discrete methods. The backward Euler method requires roughly two orders of magnitude less points than the forward Euler.

Treatment of the coupling

It is interesting to compare the performance of both the staggered and interlaced iteration methods of dealing with the coupling. Figs.3.16 and 3.17 show that the staggered solution has a poorer accuracy, as it is to be expected since the coupling equation is not being solved exactly. On the other hand, it is still stable as a result of both the electromagnetic and mechanical time stepping methods being stable.

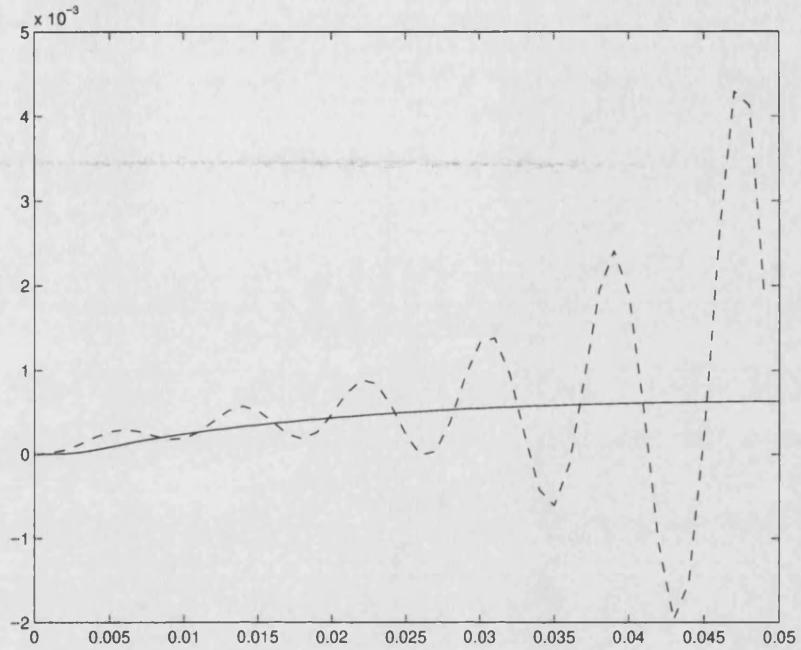


Figure 3.10: Velocity (m/s) vs. Time (s): (–) Reference, (- -) Forward Euler with $\Delta t = 10^{-3}$s.

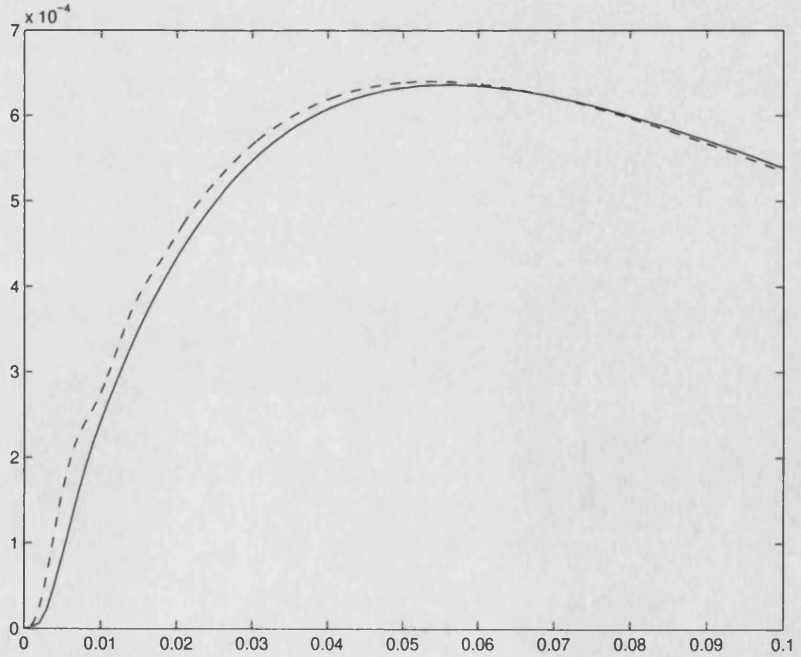


Figure 3.11: Velocity (m/s) vs. Time (s): (–) Reference, (- -) Forward Euler with $\Delta t = 10^{-4}$s.

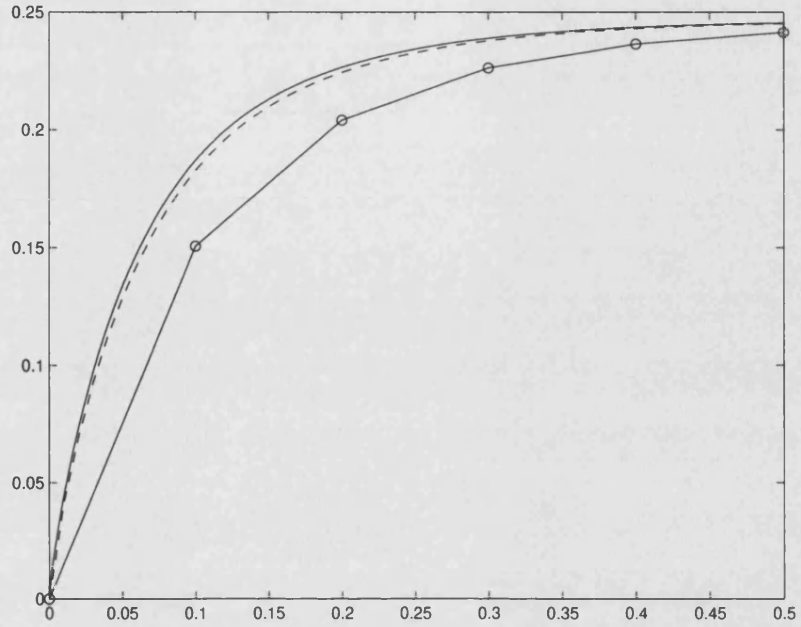


Figure 3.12: Current (A) vs. Time (s), Backward Euler: Δt = (–) 10^{-3}s, (- -) 10^{-2}s, (-o-) 10^{-1}s.

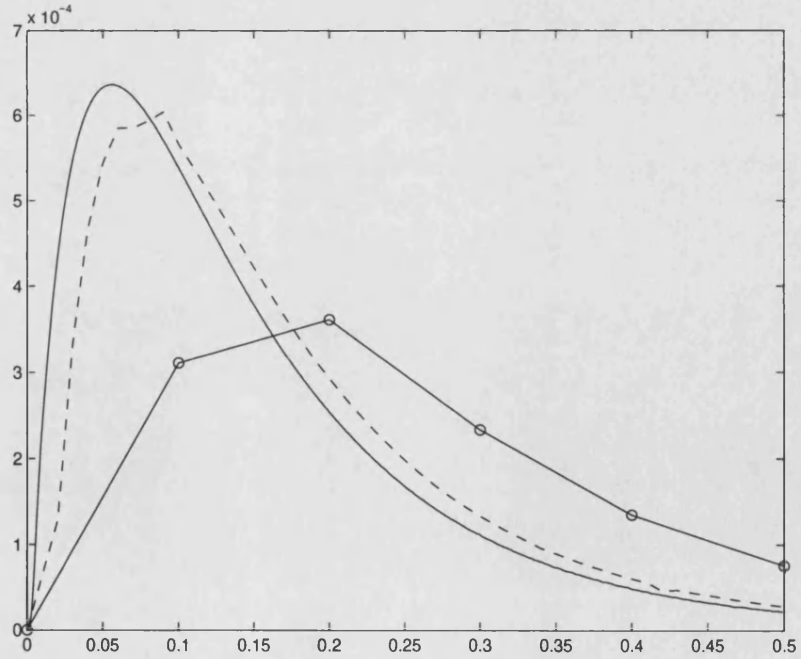


Figure 3.13: Velocity (m/s) vs. Time (s), Backward Euler: Δt = (–) 10^{-3}s, (- -) 10^{-2}s, (-o-) 10^{-1}s .

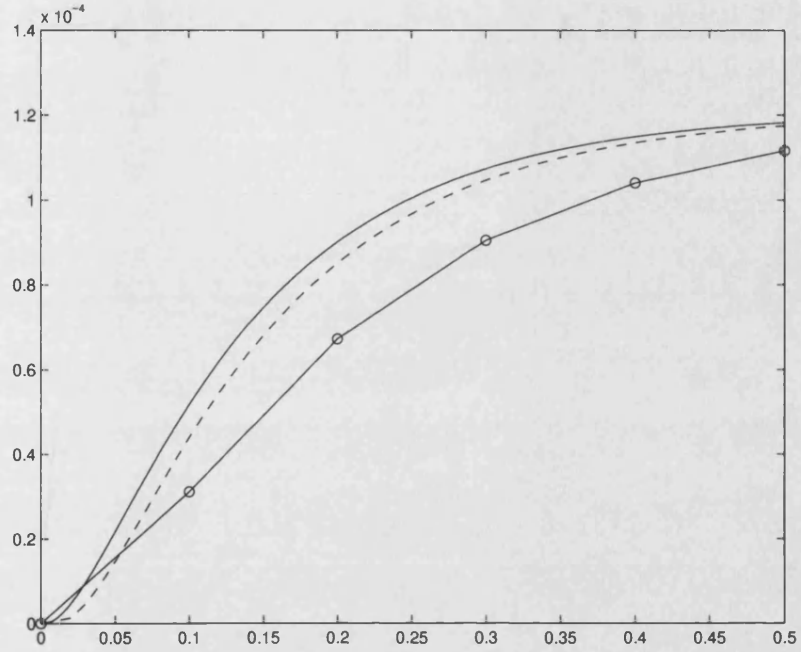


Figure 3.14: Position (m) vs. Time (s), Backward Euler: Δt = (–) 10^{-3}s, (- -) 10^{-2}s , (-o-) 10^{-1}s .

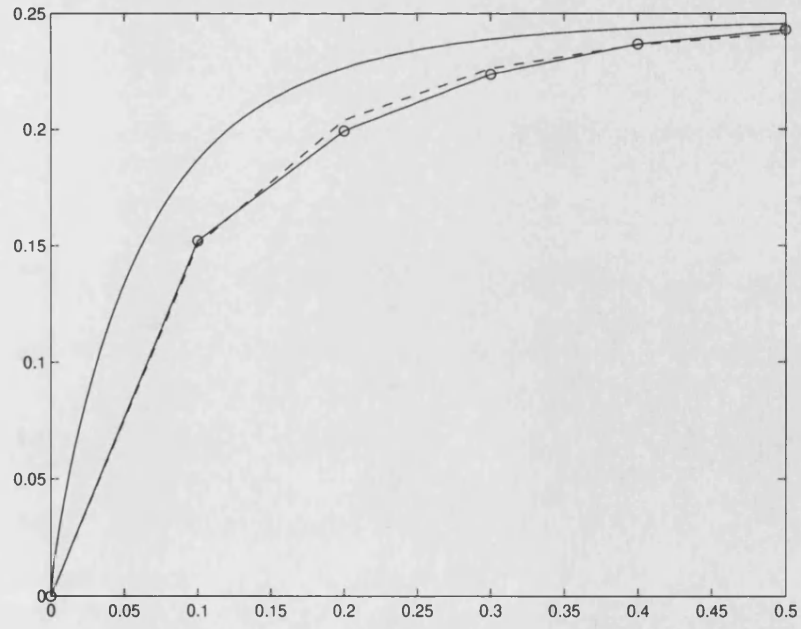


Figure 3.15: Current (A) vs. Time (s): (–)Reference; Backward Euler, $\Delta t = 10^{-1}$s with (- -)interlaced iteration, (-o-)staggered solution.

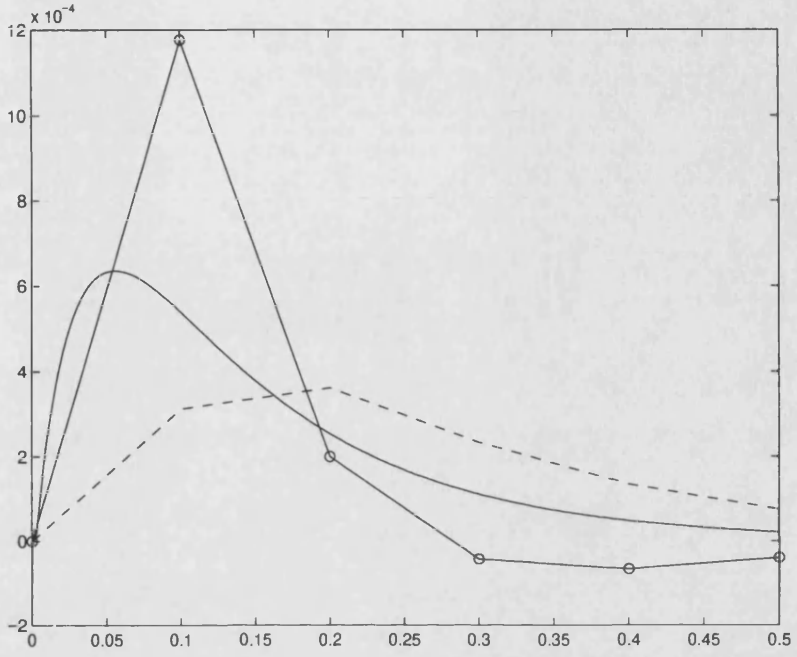


Figure 3.16: Velocity (m/s) vs. Time (s): (–)Reference; Backward Euler, $\Delta t = 10^{-1}$s with (- -)interlaced iteration, (-o-) staggered solution.

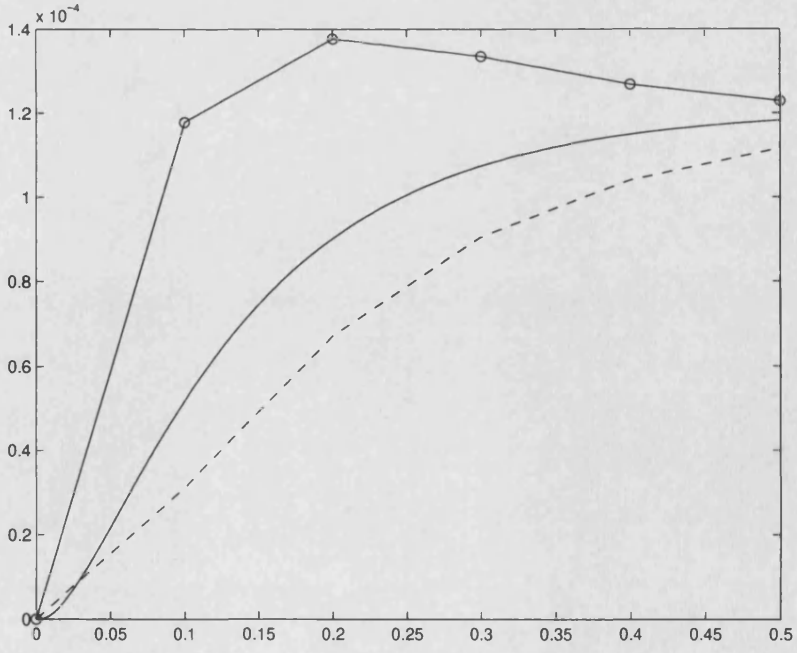


Figure 3.17: Position (m) vs. Time (s): (–)Reference; Backward Euler, $\Delta t = 10^{-1}$s with (- -)interlaced iteration, (-o-)staggered solution.

3.5.2 Comparison with measurements

In the preceding section a lumped parameter model was used as a reference for comparing results, with the confidence that the effect of the eddy current was small. In this section this assumption is justified by comparing the results of the finite element simulation against measurements taken from a prototype actuator; the dimensions, material characteristics and experimental results are reported in [45]. The closing transient treated in connection with the reduced order models was simulated again, this time using the finite element method. The staggered and interlaced methods were used.

The simulated transient consists in the closing of the armature loaded only by its own weight; the initial position is 0.0mm and the final position is 0.73mm. Fig.3.18 compares the measured and calculated currents with a time step Δt of 10^{-2}s for both the staggered and the interlaced iteration; Fig. 3.19 shows the corresponding positions. The interval of armature movement is approximately 0.04 to 0.07s; the electrical transient persists until about $t = 0.5$s (not shown entirely). It can be appreciated that the interlaced process gives better results, as was to be expected since the coupling equation is solved consistently.

If a shorter time step is used, both the interlaced and the staggered solutions get closer to the measured values and very close to each other. For $\Delta t = 10^{-3}$s, the currents are practically the same as seen in Fig.3.20. A slight deviation in the position is observed in Fig.3.21. The difference with respect to the measured values at the instant of closing is attributed to the elastic impact not being modeled.

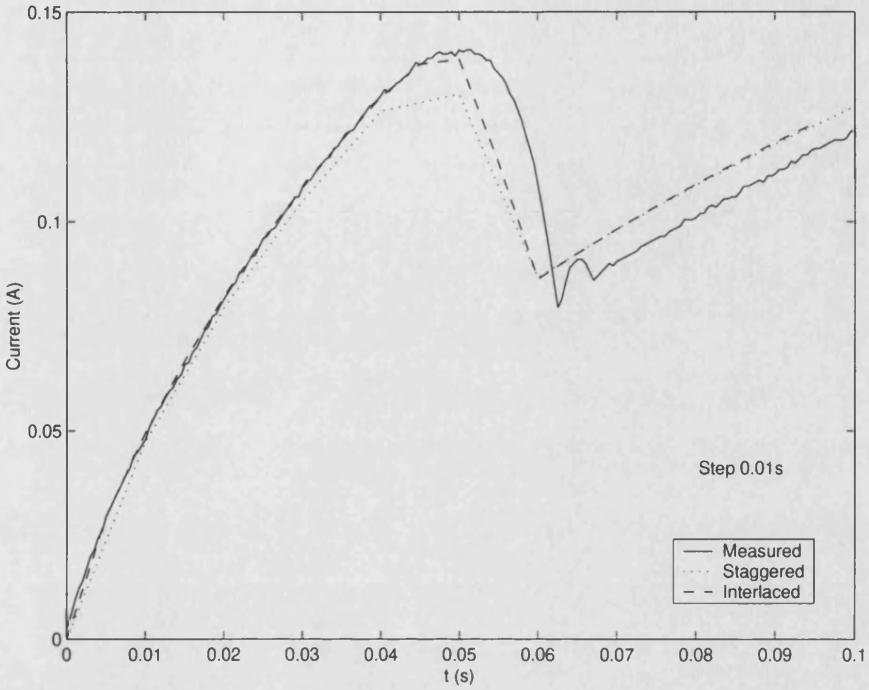


Figure 3.18: Measured and simulated current with $\Delta t = 10^{-2}$s.

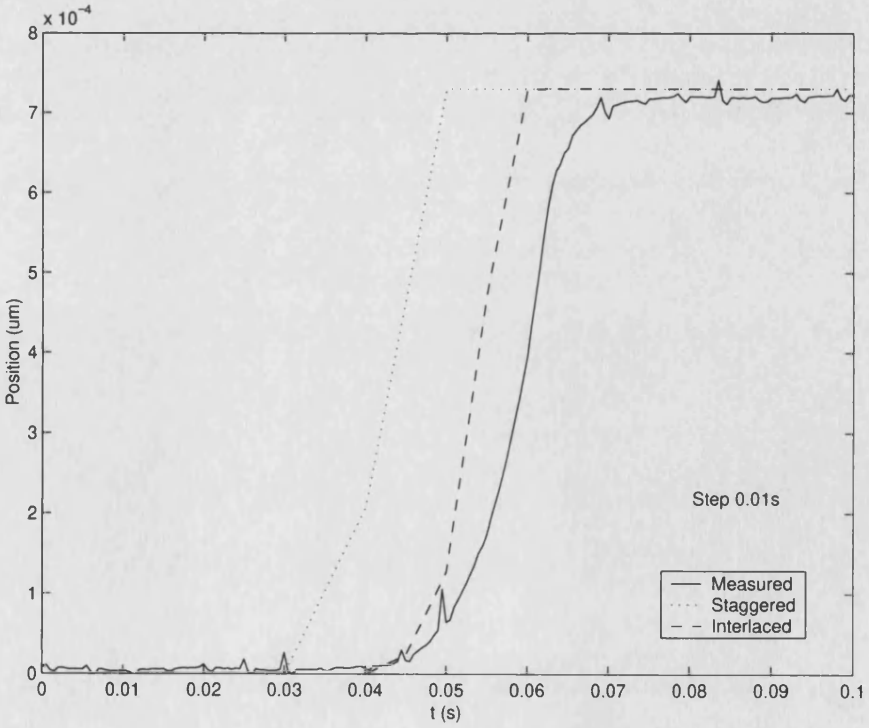


Figure 3.19: Measured and simulated position with $\Delta t = 10^{-2}$s.

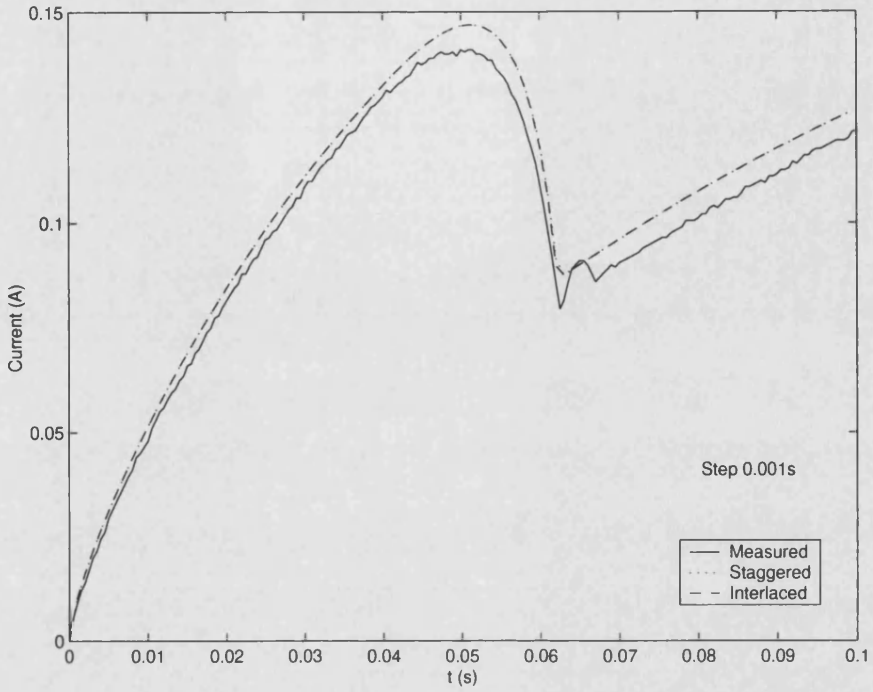


Figure 3.20: Measured and simulated current with $\Delta t = 10^{-3}$s.

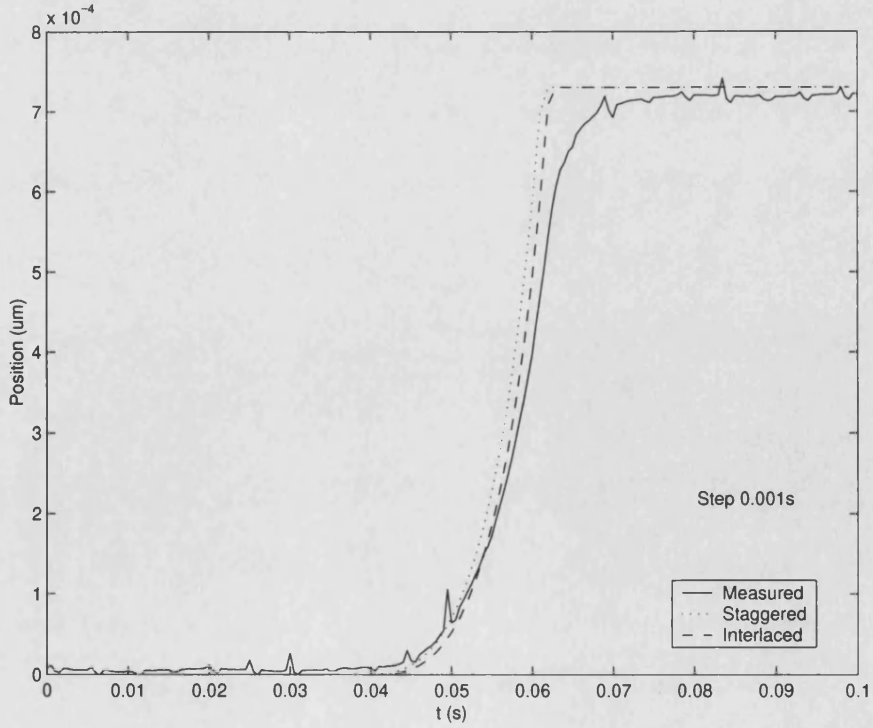


Figure 3.21: Measured and simulated position with $\Delta t = 10^{-3}$s.

3.5.3 Induction motor

The switch-on transient of the solid rotor induction motor outlined in Fig. 3.22 was simulated for the no load case. The staggered and interlaced schemes were compared; the mechanical equation was solved using the modified Euler and backward Euler methods. First, a very small time step (0.5ms) is used to verify the implementations; as seen in Fig. 3.23, all methods predict the same velocity. With a time step of 1ms, the responses differ; the staggered–backward Euler and the interlaced–backward Euler schemes agree with each other but predict a larger speed than they should and the staggered–modified Euler scheme gives an even bigger velocity. These effects may be attributed to the different accuracy characteristics of the mechanical equation solutions, which begin to be noticeable for this time step size. The mechanical time constant is important to consider too: in the case under study, the mechanical response is swift so that the differences may as well be attributed to the stability characteristics of the methods. This is confirmed when a system with slower response is considered; with a mechanical load of ten times the moment of inertia of the rotor alone, the responses agree again, as seen in Fig. 3.25.

3.5.4 Rotational test rig

In this section we simulate the dynamic response (with movement) of a rotational test rig previously used for static tests [70, 71]. The test rig was designed to validate 3D nonlinear time-transient codes; it has a configuration similar to that of a switched reluctance machine made of solid medium-carbon steel and mounted in a nonmagnetic cage which can rotate about a stainless steel shaft. The device is shown in Fig. 3.26.

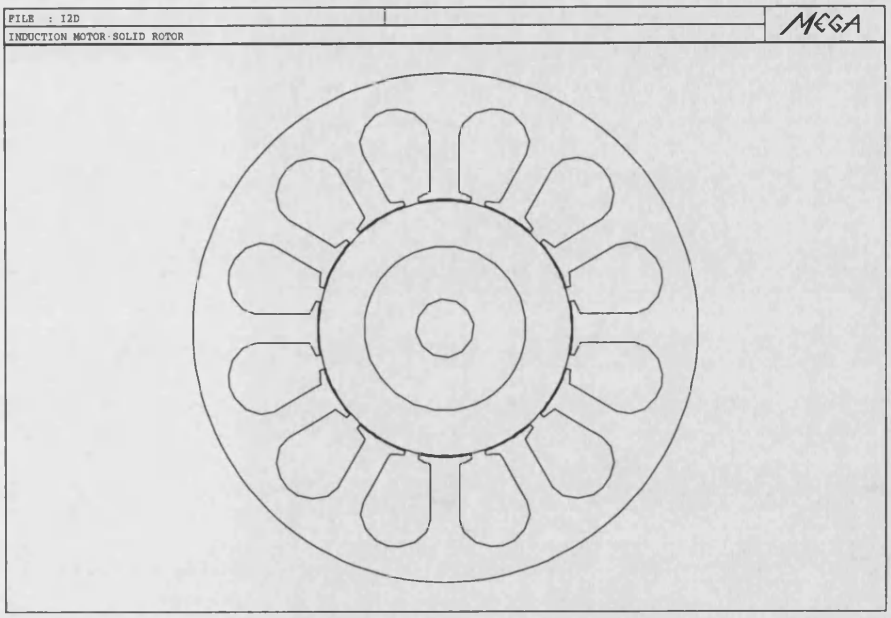


Figure 3.22: Solid rotor induction motor.

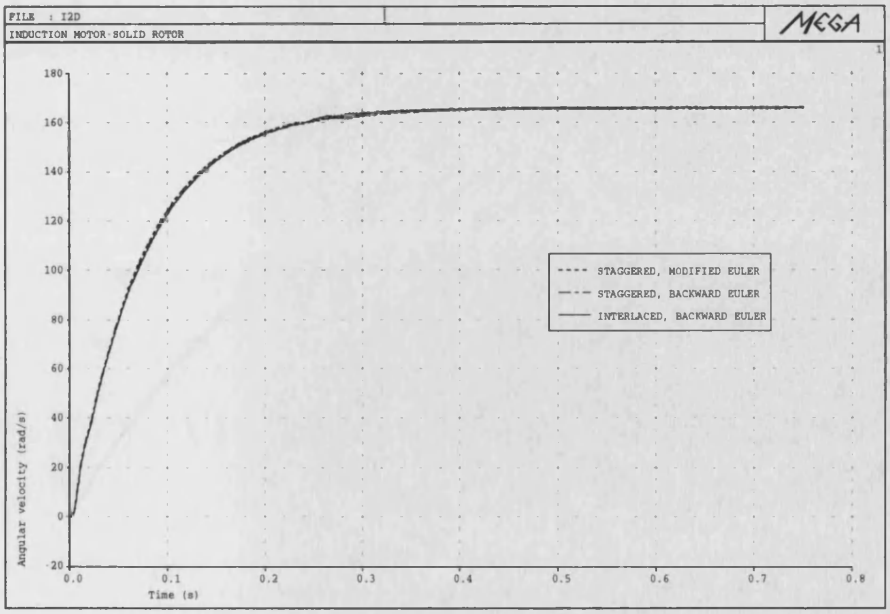


Figure 3.23: Rotor angular velocity. $\Delta t = 0.5$ms

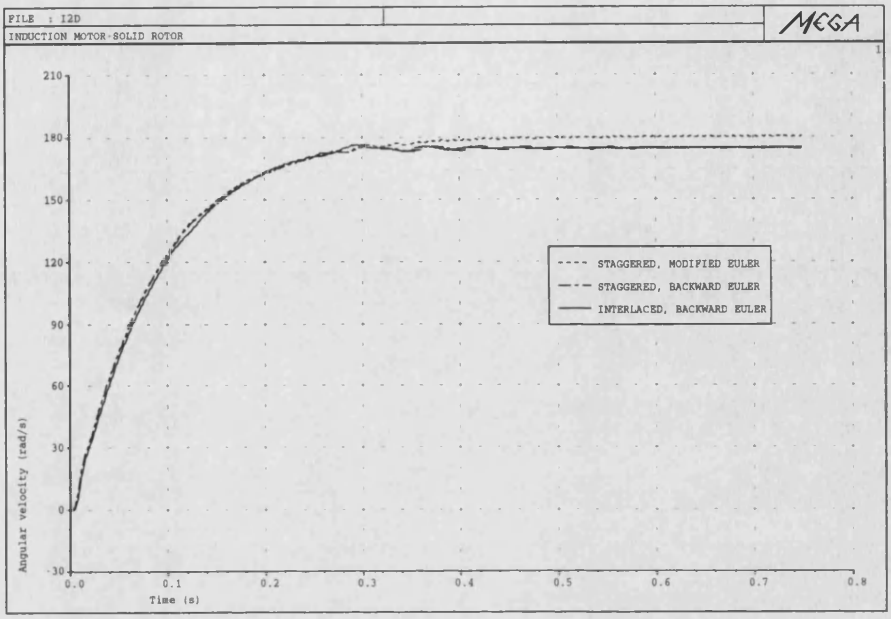


Figure 3.24: Rotor angular velocity. $\Delta t = 1$ms

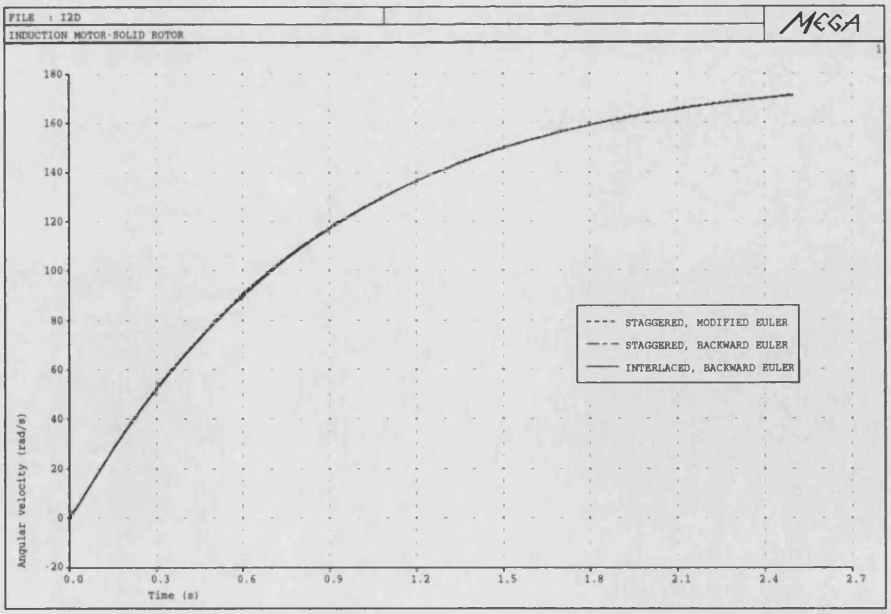


Figure 3.25: Rotor angular velocity, large inertia. $\Delta t = 1$ms

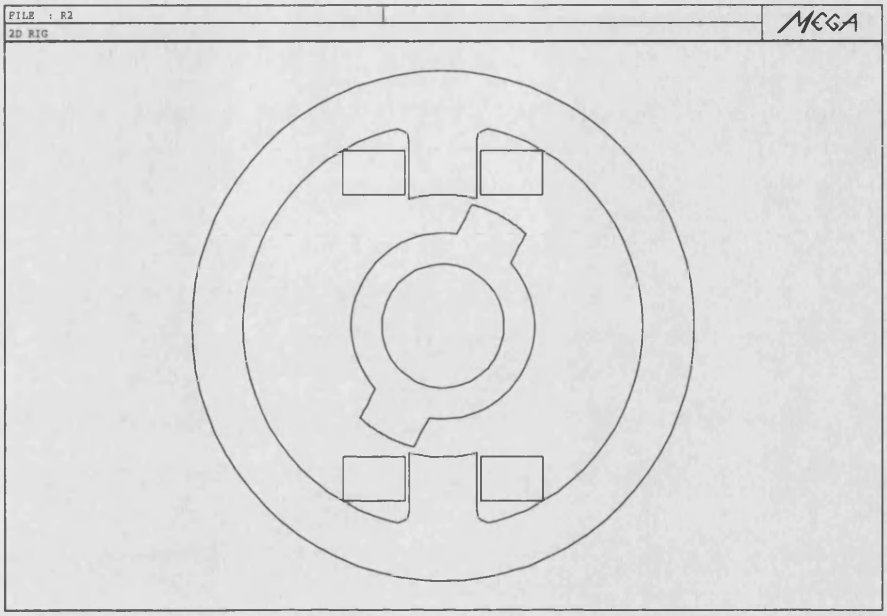


Figure 3.26: Rotational test rig.

The initial position of the rotor is 23 degrees out of alignment with the stator. When the coils are energized by a step voltage source, the rotor tends to align with the stator but it does so following an oscillating movement until reaching the final position. We will compare the measured and calculated position and current as functions of time. For the simulation the staggered and interlaced schemes are used, while the mechanical equation is solved by the modified and backward Euler formulas. Thus, a particular method is identified by giving the treatment of the coupling and the formula for the mechanical equation, for instance staggered–backward Euler. Each method is considered in turn. A two-dimensional model was used in the simulations since good agreement with the measurements is observed, and the fundamental characteristics of the distinct schemes can be appreciated readily while avoiding the lengthy three-dimensional calculations.

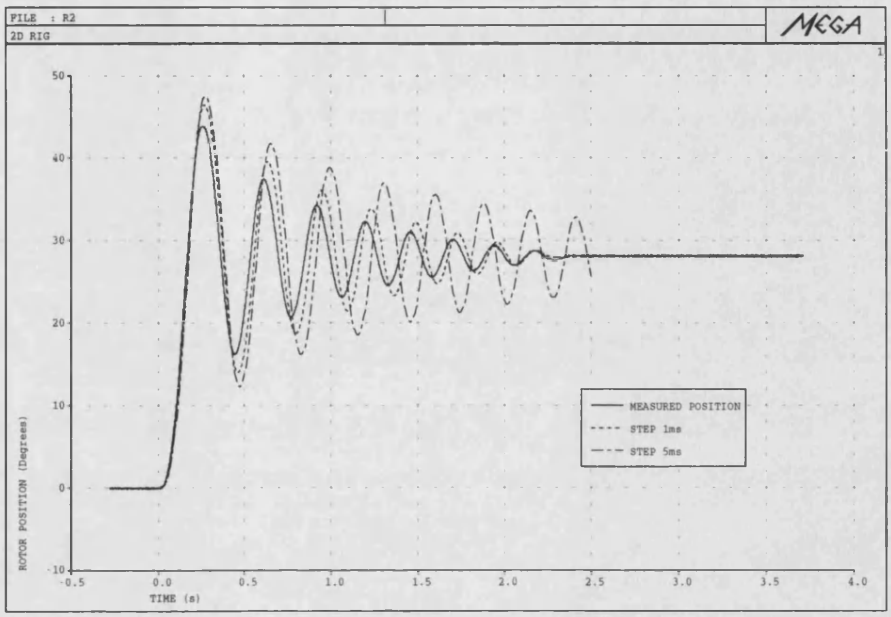


Figure 3.27: Transient position for staggered–modified Euler method with small time steps.

Staggered–Modified Euler

The staggered-modified Euler scheme gives the results plotted in Figs.3.27 and 3.28. For a small time step (1ms), the results are reasonably close to the measurement; however, larger time steps result in the scheme becoming unstable. For a time step of 5ms, the oscillations take much more time to disappear and the phase correspondence, which for 1ms was approximately achieved, is lost. For a time step of 25ms the oscillations are not damped at all and the behavior is that of a lossless system; this is physically impossible since explicit losses are included in the model, namely mechanical friction and eddy currents. Consequently, the source of the error is attributable to a instability on the part of the mechanical time stepping formula, the modified Euler method, as confirmed dramatically for time steps sizes of 50 and 100ms, Fig. 3.28.

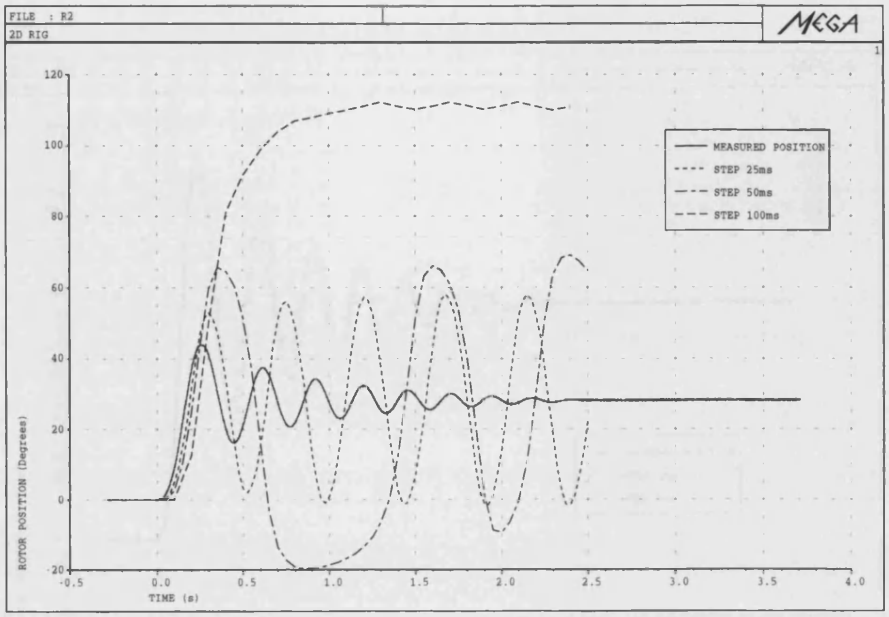


Figure 3.28: Transient position for staggered–modified Euler method with large time steps.

Staggered–Backward Euler

The results for this scheme are plotted in Figs.3.29 and 3.30. The stability characteristics are much better in this case: the results are essentially unchanged for time steps of up to 50ms. Moreover, the results for 1 and 5ms are almost identical, meaning that a time step of 5ms provides a converged result. A step size of 25ms still provides a good answer, while with 50ms the response shows the time step corners. Finally, for a step size of 100ms the scheme losses stability.

Interlaced–Backward Euler

This scheme gives the results shown in Figs.3.31 and 3.32. Several characteristics of the method can be appreciated. First, with a time step of 1ms, the results are very good; however, for larger time steps, the scheme seems to add a numerical damping

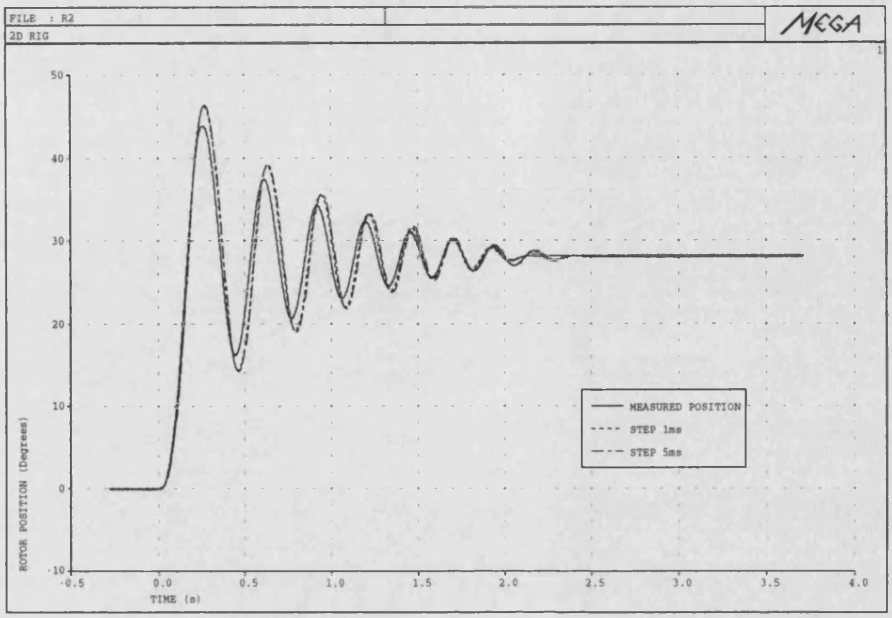


Figure 3.29: Transient position for staggered–backward Euler method with small time steps.

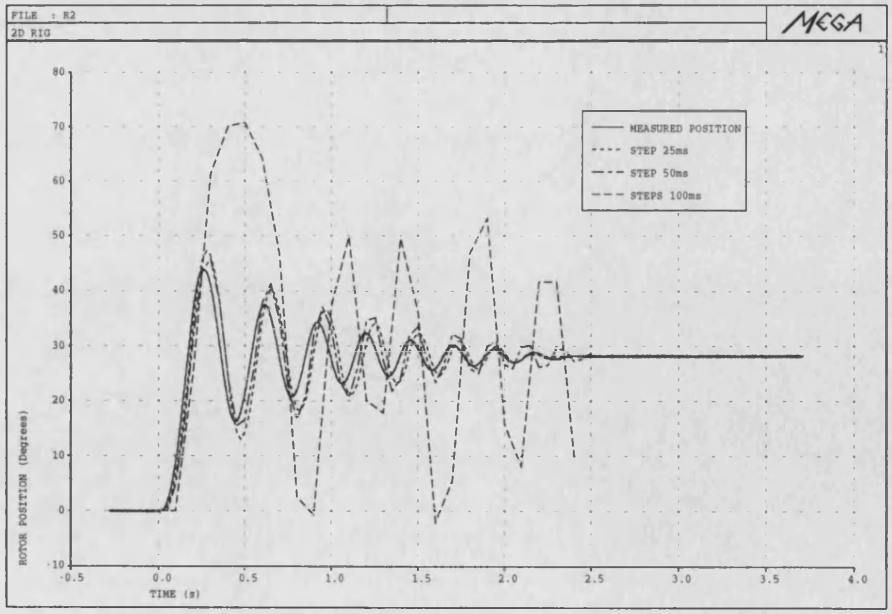


Figure 3.30: Transient position for staggered–backward Euler method with large time steps.

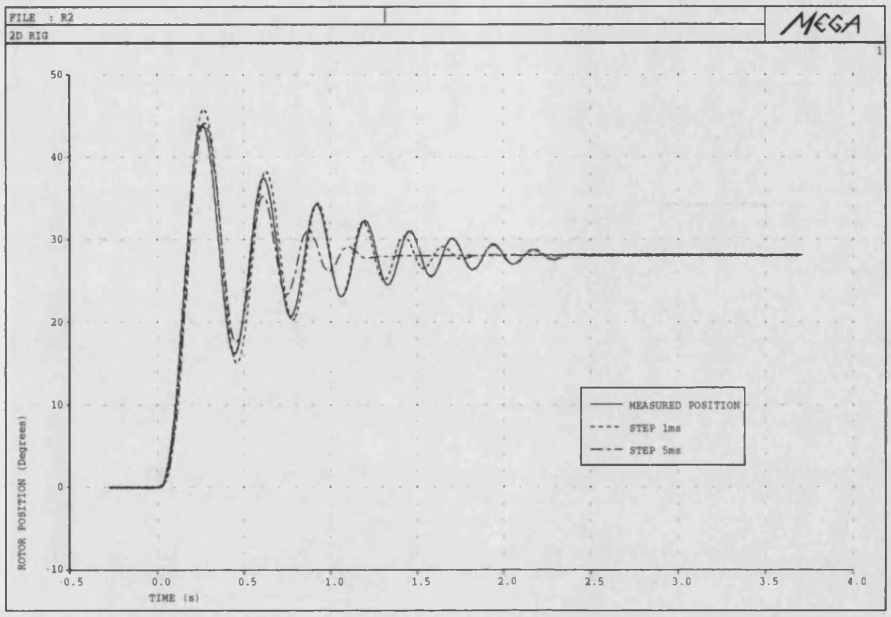


Figure 3.31: Transient position for interlaced–backward Euler method with small time steps.

to the system; the larger the time step, the bigger the damping. Interestingly, larger time steps are not possible since they lead to a non convergent solution.

Comparison of methods

The responses for the three schemes with a time step of 1ms are shown in Figs.3.33 and 3.34.

It has been observed that the modified Euler formula is unstable; this is not surprising since it uses a forward (explicit) expression for the derivatives. The interlaced iteration introduces excessive damping for large time steps in this case (but not for non-oscillatory cases); on the bright side, it provides a safety mechanism since excessively large time steps are avoided (the algorithm fails to converge).

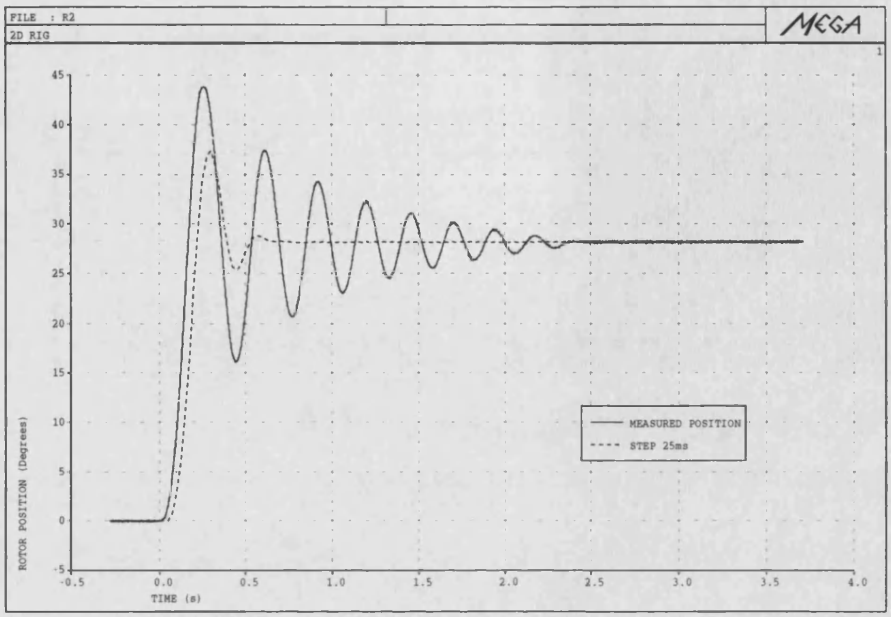


Figure 3.32: Transient position for interlaced–backward Euler method with large time steps.

The staggered-backward Euler scheme delivers a better performance than the other methods. It shows good stability only losing it for very large time steps; its accuracy is very good and is less affected by the step size. As a consequence, essentially the same result is obtained with a moderately large time step than with a small time step. The response can be characterized quickly with the confidence that the results are close to the best obtainable.

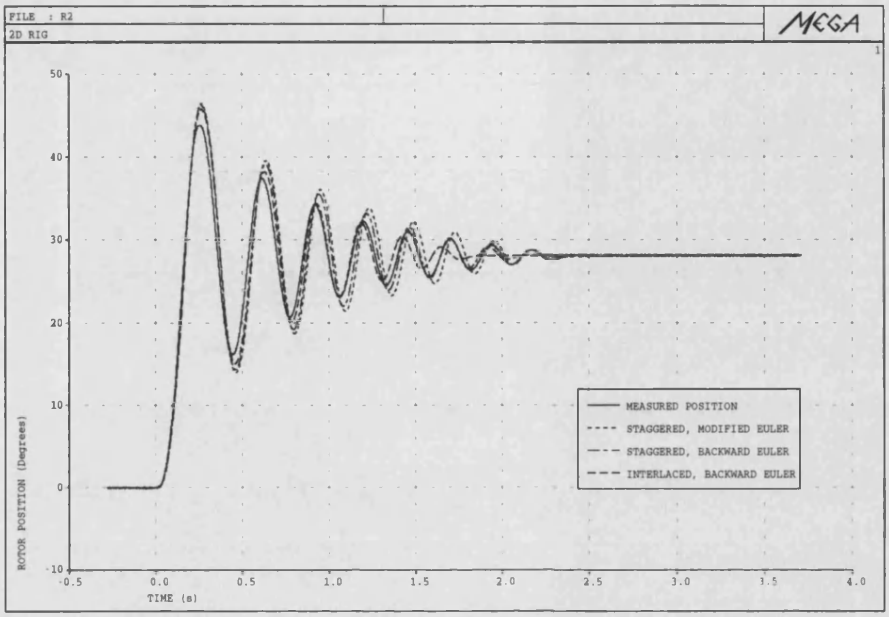


Figure 3.33: Transient position with Δt =1ms.

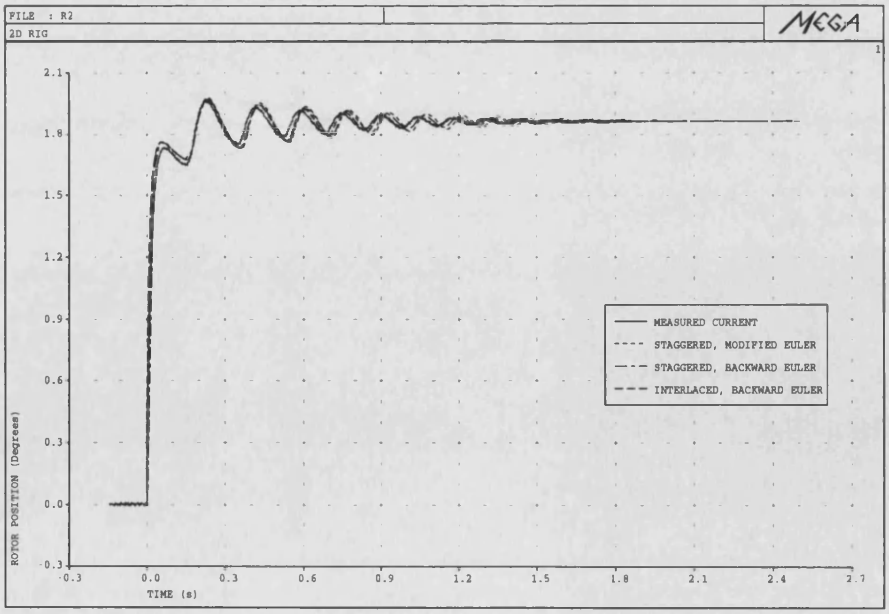


Figure 3.34: Transient current with Δt =1ms.

3.6 Conclusions

Several approaches for the solution of coupled problems have been considered. These may be applied to both static and dynamic problems. Some approaches commonly used have been reviewed and two new concepts introduced; these are the use of coupling variables and the interlacing of nested iterations.

The use of coupling variables to steer the solution of the coupled problem has the advantage that existing single-domain solvers can be utilized, albeit only after some extensions are made. The point is that variables pertaining to other domains are not mixed in the same solver. This can lead to a more logical layer of solvers, where high-level solvers like circuit simulators use the results of low-lever solvers like finite element codes.

The solution of the coupled problem reduces to the solution of a nonlinear problem. The interlaced iteration approach aims at achieving a fast convergence (comparable to the convergence of a full Newton iteration), by solving separate domains. It is then very well suited to cases where separate domain solvers exist. It can theoretically be applied to both static and dynamic problems, but our experience has shown that it is best suited to static problems; for dynamic problems, the method introduces a numerical damping in the case of oscillatory movement (although it seems to have no problems in cases where the position increases or decreases monotonically).

We have shown that for dynamic problems a strong coupling is often not required. The staggered solution provides satisfactory results most of the time; the reason for this is that the time constant of the electromagnetic domain is much smaller then the time constant of the mechanical domain. The only consideration is thus to

ensure that a stable time-stepping formula is applied in each separate domain. In the case of oscillating systems, the response may be better approximated by using lower order formulas; the response calculated by using higher order formulas appears to require smaller time steps to show equivalent precision. This is because the multistep formulas use past states and are thus slow to reflect the changing conditions.

Chapter 4

Magneto-elastic problems

In this chapter we extend the ideas developed in the treatment of rigid-body systems, to deal with cases where there is an elastic deformation of the body caused by a magnetic field. We limit our interest to those cases where the magnetostrictive effect is so small that it can be ignored and where the force responsible for the deformation is the surface magnetic pressure. The subject has received less attention than the study of the deformation due to body Lorentz forces (e.g. eddy currents in a plate in a magnetic field).

The discussion is organized as follows. First, the force distribution due to the magnetic field is analyzed and then an application problem with the required characteristics is proposed. A linear approximation is made and some preliminary evaluations developed. A static field solution considering the magnetic material nonlinearity is presented, along with some concerns about the use of line search algorithms. The solution of the coupled problem using the Gauss–Seidel iteration is included. Finally, a comparison between measured and calculated results is made.

4.1 Stress due to magnetic field

The distribution of force in a ferromagnetic elastic material in the presence of a magnetic field has been treated by a number of authors, but some discrepancies have arisen. The various expressions for the stress differ in the magnitude of the local stress but agree on the total force exerted on a volume [6]. It is now known that a general formulation requires taking into account the electrodynamic, mechanical and thermodynamic aspects of the problem [72].

For a linear magnetic material, the stress tensor can be written [73]:

$$T_{mn} = \mu H_m H_n - \frac{1}{2}\delta_{mn} H_k H_k \left(\mu - \frac{\partial \mu}{\partial \rho}\rho\right), \tag{4.1}$$

where ρ is the density. If the magnetostrictive effect is small, the stress tensor is simply:

$$T_{mn} = \mu H_m H_n - \frac{1}{2}\delta_{mn}\mu H_k H_k, \tag{4.2}$$

which is Maxwell's stress tensor except for the permeability μ [6]. We can make use of this analogy in the following form: for an elastic body such that the magnetostrictive effect is small, we can specify a surface in the surrounding air in the usual way, and make it shrink so that the integration is carried arbitrarily close to the ferromagnetic material. We are not interested in the total force; instead, we use the stress tensor (4.2) with $\mu = \mu_0$ to calculate the traction (surface force):

$$\tau_m = T_{mn} n_n, \tag{4.3}$$

where n_n is a component of the unitary vector normal to the surface of the material.

Thus, for structures where the magnetic surface force accounts for the deformation, the problem may be treated as a coupled field–structure problem, also known as Type-I coupled problem [62]. Interestingly, consideration of the local force inside the ferromagnetic material leads to the same conclusion [6]. In this case, the mechanical boundary conditions at the interface indicate that the surface force dominates over the magnetostrictive force, so that the latter may be ignored.

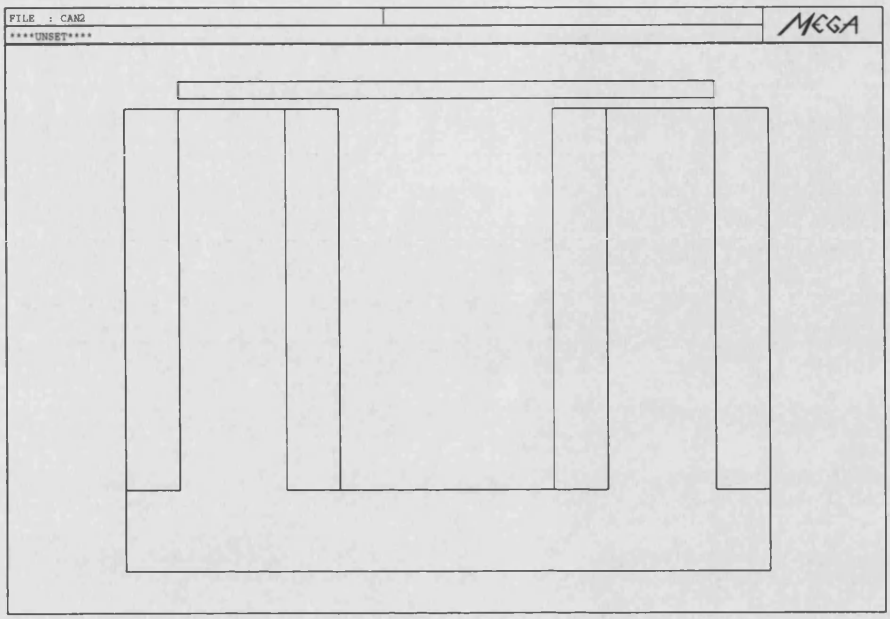


Figure 4.1: Magnet and plate.

4.2 A plate problem

The following problem is proposed. A ferromagnetic plate serving as a flux path is suspended close to the pole faces of a magnet and fixed at the middle or at one of the extremes, Fig. 4.1. The thin plate will be saturated, but the flux entering the plate from below will probably exceed the flow leaving it above, thus creating a force tending to deform the plate and make it touch the magnet pole. We want to calculate the deformation as a result of a given current.

4.2.1 Magnetic circuit approach

We solve first a simplified form of the problem of Fig. 4.1 using a linear magnetic circuit. Since the permeability of the ferromagnetic material is much higher than that of the air, we can make the approximation $\mu \to \infty$ and use the integral form

of Ampere's law:

$$\oint_L \mathbf{H} \cdot d\mathbf{L} = Ni \tag{4.4}$$

to solve for the magnetic flux density B:

$$B = \frac{\mu_0 Ni}{g}, \tag{4.5}$$

where N is the number of turns in the coil, i the current in the coil and g the air gap length, in this case the distance separating the plate and the magnet pole face. We can then consider the cantilever as a beam and pose a strength-of-materials problem as in Fig. 4.2, where a, b and L are lengths and q is the applied force per unit length. This force density can be calculated starting from (4.2) and setting $\mu = \mu_0$. In matrix form, this tensor is:

$$\mathbf{T} = \begin{bmatrix} \frac{B_1 B_1}{\mu_0} - \frac{B^2}{2\mu_0} & \frac{B_1 B_2}{\mu_0} & \frac{B_1 B_3}{\mu_0} \\ \frac{B_2 B_1}{\mu_0} & \frac{B_2 B_2}{\mu_0} - \frac{B^2}{2\mu_0} & \frac{B_2 B_3}{\mu_0} \\ \frac{B_3 B_1}{\mu_0} & \frac{B_3 B_2}{\mu_0} & \frac{B_3 B_3}{\mu_0} - \frac{B^2}{2\mu_0} \end{bmatrix} \tag{4.6}$$

For the idealized case considered in this section, the flux density has components $B_1 = 0$, $B_2 = B$, $B_3 = 0$, while the surface vector has components $n_1 = 0$, $n_2 = -1$, $n_3 = 0$. Thus, according to (4.3), the components of the traction are:

$$\begin{cases} \tau_1 = \frac{B_1 B_2}{\mu_0} = 0 \\ \tau_1 = -\frac{B_2 B_2}{\mu_0} + \frac{B^2}{2\mu_0} = -\frac{B^2}{2\mu_0} \\ \tau_3 = \frac{B_3 B_2}{\mu_0} = 0 \\ . \end{cases} \tag{4.7}$$

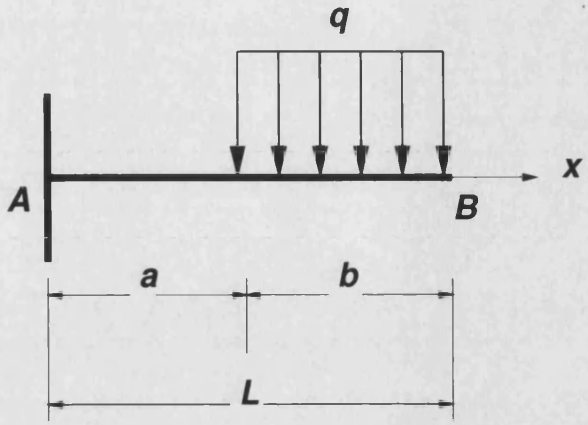


Figure 4.2: Strength-of-materials beam problem.

The total force is given by the surface integration of the traction:

$$f = -\int_S \frac{B^2}{2\mu_0} dS, \tag{4.8}$$

so that the force density is:

$$q = \frac{dB^2}{2\mu_0} = \frac{\mu_0 d}{2}\left(\frac{Ni}{g}\right)^2 \tag{4.9}$$

where d is the width of the plate and the expression for the flux density (4.5) has been replaced.

The general solution to the beam problem is given in [74]. The vertical displacement is:

$$v(x) = \begin{cases} -\frac{qbx^2}{12EI}\left(3L + 3a - 2x\right) & 0 \le x \le a \\ -\frac{q}{24EI}\left(x^4 - 4Lx^3 + 6L^2x^2 - 4a^3x + a^4\right) & a \le x \le L. \end{cases} \tag{4.10}$$

E is Young's modulus for the elastic material, and I is the moment of inertia which

a	0.02m
b	0.02m
d	0.02m
h	0.003m
g	0.001m

Table 4.1: Problem dimensions.

for a plate of height h and width d is given by:

$$I = \frac{dh^3}{12}. \tag{4.11}$$

The displacement at the tip ($x = L$) is:

$$\delta_B = \frac{q}{24EI}\left(3L^4 - 4a^3L + a^4\right). \tag{4.12}$$

We use these equations to gain an understanding of the magnitude of the quantities involved. Assuming as constants the Young's modulus E, the width d and lengths a, b and L, we are free to vary the height of the plate h, the air gap length g and the magnetomotive force Ni to produce a suitable displacement δ_B at the tip. A large displacement is desirable in order to allow the measurement of the displacement. A small height h will produce a large deflection, but the saturation will complicate the numerical solution.

For the dimensions of Table 4.1, the calculated displacement is shown in Fig. 4.3 for several air gap lengths.

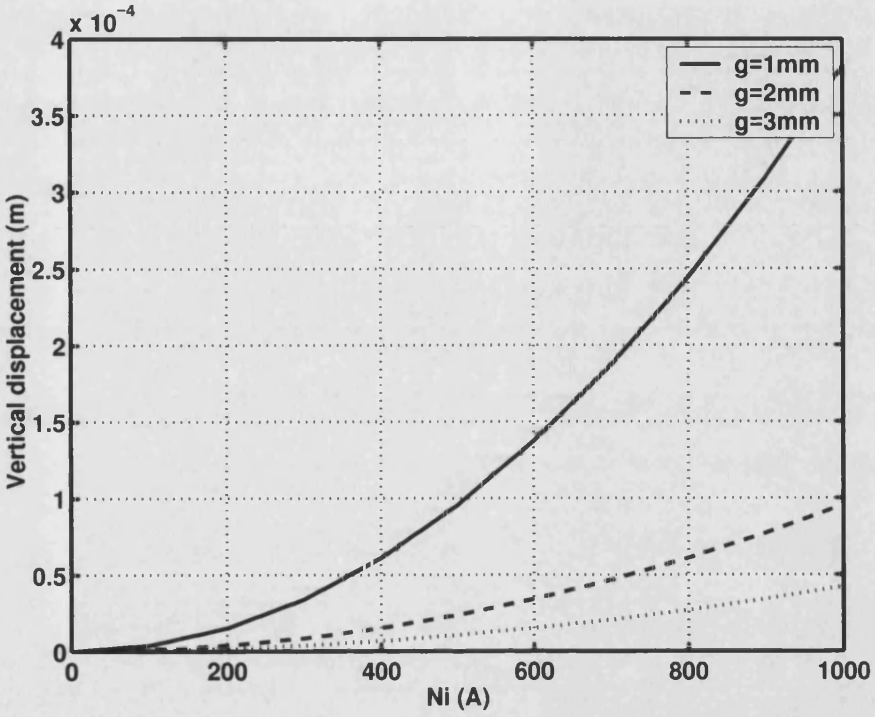


Figure 4.3: Vertical displacement at the tip, δ_B.

4.3 Magnetic field solution

The static magnetic problem was solved using the finite element method. For $Ni = 500$ and $g = 2mm$, the number of iterations required to solve the nonlinear magnetostatic problem was 6, showing some saturation at the plate. The flux lines are shown in Fig. 4.4; an artificial boundary was added to terminate the air region and this can be appreciated in the distorted shape of the flux line at the top of the model. We assume that this boundary has a negligible effect on the results (force).

As explained previously, the Maxwell stress tensor is used to calculate the surface force exerted by the magnetic field over the plate. Since the normal surface vector is y-directed in this case, the B_y component of the magnetic flux density vector

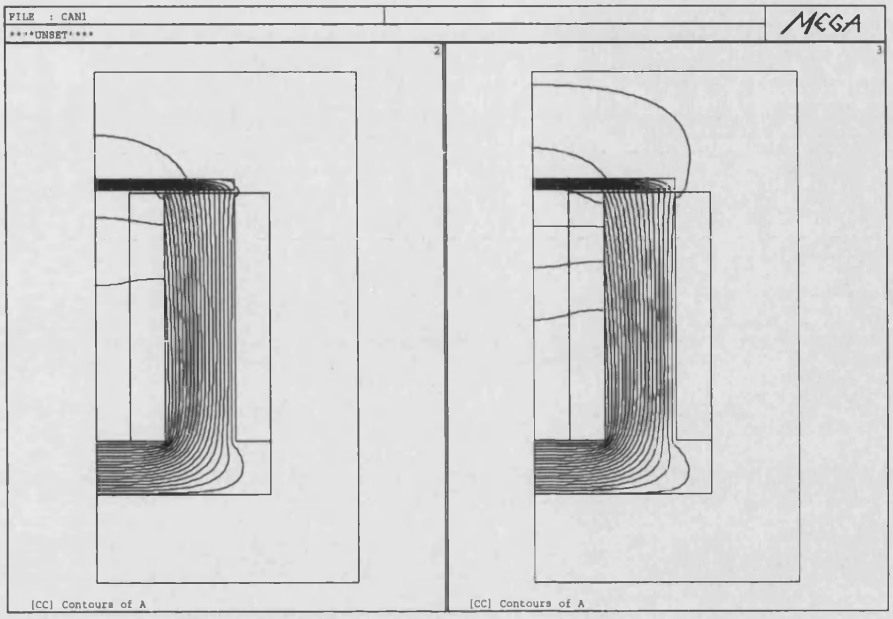


Figure 4.4: Flux plot for $Ni = 500$A and 1000A, $g = 2$mm.

determines the magnitude of the initial vertical force, according to (4.3).

From the finite element static solution, the B_y component is as shown in Fig. 4.5. The element numbers in this figure refer to the sequence illustrated in Fig. 4.6. The shape of the force profile is as expected: very close to a rectangle between elements 11 and 20, where most of the flux is concentrated. The distribution is not uniform, however, and there is some flux that passes through the adjacent elements. Also, there is a small leakage component on the upper side of the plate, elements 26 to 45. On the other hand, the magnitude of the flux density is not quite as predicted by the linear magnetic circuit model, which gives a value of 0.62T. The difference arises because the circuit model does not include the reluctance of the nonlinear material.

It is interesting to compare the final flux density profile with those obtained for subsequent iterations in the Newton solution of the nonlinear magnetostatic problem. Fig. 4.7 compares the final and first iteration flux density profile with $Ni = 500A$

and $g = 2mm$. They are almost the same and so will be the force distribution; this is important if the coupled problem is to be solved by interlacing. For a smaller air gap, the magnetic material is more saturated and the results for successive iterations show a greater variation, Fig. 4.8. In fact, the first iterations show a larger value of flux than the final one, which could result in the interlaced iteration becoming unstable (because the deflection will be overestimated in the first correction, leading to increased flux density and so on).

Closer examination of the Newton iteration for the solution of the magnetostatic problem reveals the cause of the overestimation of the flux. The particular solver used for this case was one which takes full Newton steps, that is, the correction vector $\Delta\mathbf{x}$ in the equation $\mathbf{J}\Delta\mathbf{x} = -\mathbf{F}$ is used to update the states ($\mathbf{J}$ is the Jacobian matrix); this can be appreciated in the convergence history shown in Table 4.2, where the second and third solutions have larger residuals than the first one. We see that taking full Newton steps always is not a good strategy for several reasons: 1)for heavily saturated problems, the algorithm will fail, and 2)for the coupled magneto-elastic problem, the interlaced iteration could fail.

A corrected algorithm must be used instead, this time with a line search which eliminates steps leading to residuals greater than the current one. When implemented, this strategy results in the flux density from initial iterations being always smaller than the final, as seen in Fig. 4.9. This is very convenient since the position will be updated without the risk of falling in the unstable region. The convergence history is shown in Table 4.3.

Even for very large currents, the described algorithm results in the flux density approaching its true value from below, as illustrated in Fig. 4.10. Note how in this case, with $Ni = 5000$A, the flux will tend to escape from the upper side of the plate,

Iteration	$\|\Delta \mathbf{x}\|$	$\|\mathbf{F}\|$
1	1.6897×10^{-1}	50.345
2	1.1695×10^{-2}	251.93
3	8.7740×10^{-3}	86.324
4	3.0262×10^{-3}	18.324
5	2.6838×10^{-4}	1.3584
6	2.0026×10^{-6}	1.0018
7	1.3892×10^{-10}	1.8093×10^{-6}

Table 4.2: Convergence history, $Ni = 500$A, $g = 1$mm, full Newton steps.

Iteration	$\|\Delta \mathbf{x}\|$	$\|\mathbf{F}\|$
1	1.6897×10^{-1}	50.345
2	2.6314×10^{-2}	12.703
3	5.5873×10^{-3}	12.100
4	1.6612×10^{-4}	9.7771×10^{-1}
5	9.2388×10^{-7}	1.2415×10^{-2}
6	1.5300×10^{-10}	5.7180×10^{-6}

Table 4.3: Convergence history, $Ni = 500$A, $g = 1$mm, line search steps.

resulting in a somewhat reduced net force. The force distribution in the saturated case becomes even more complicated due to the augmented magnitude of the x component, as illustrated by Figures 4.11 and 4.12.

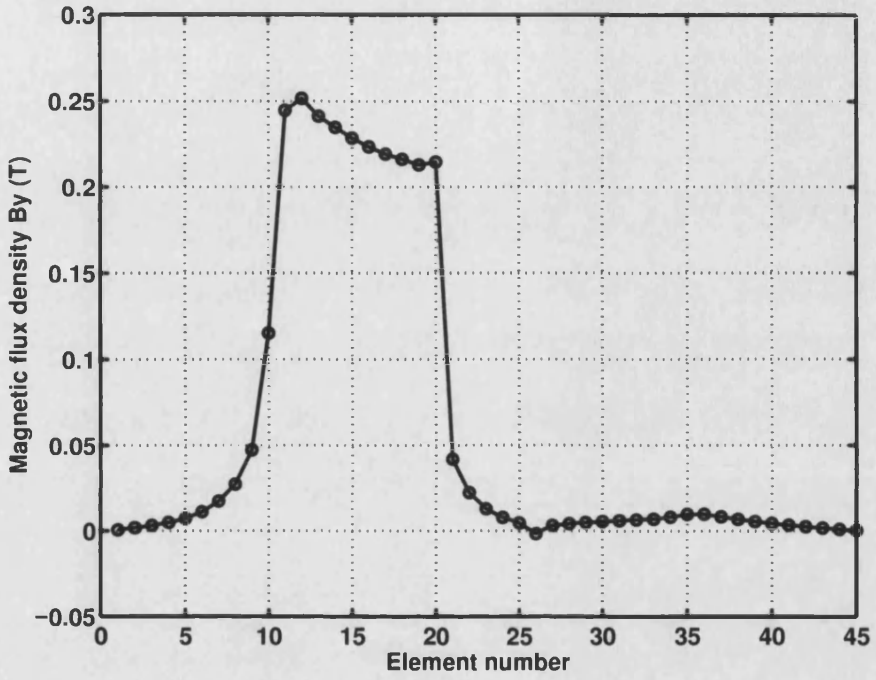


Figure 4.5: B_y alongside the plate for $Ni = 500\text{A}$ and $g = 1\text{mm}$.

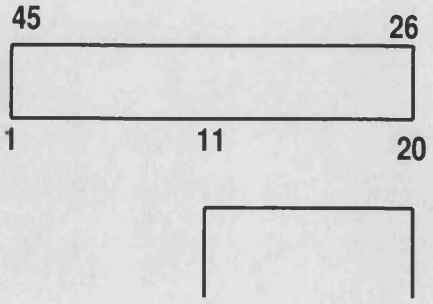


Figure 4.6: Element numbers along plate.

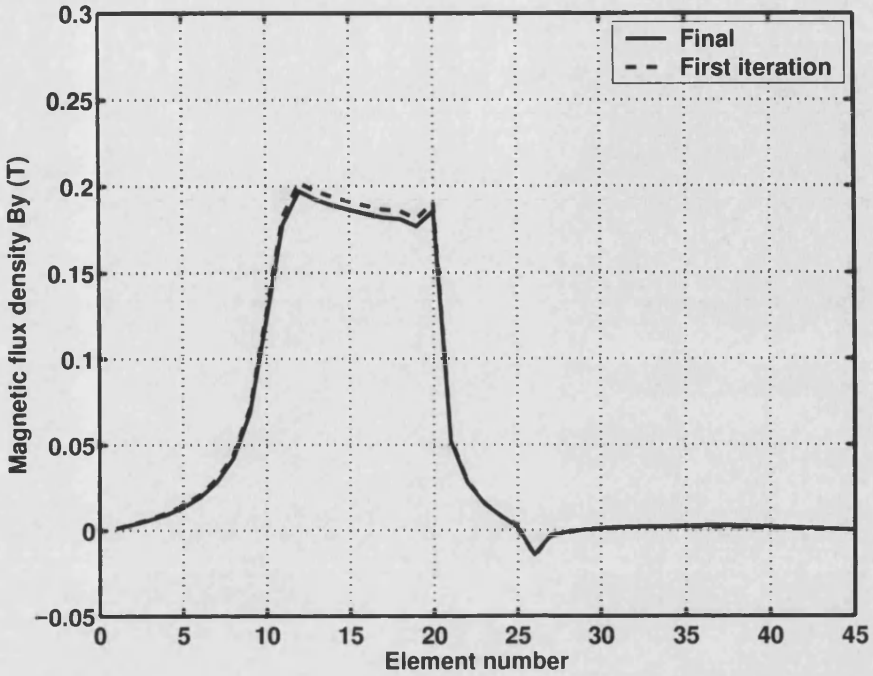


Figure 4.7: B_y for Newton iterations, $Ni = 500\text{A}$, $g = 2mm$.

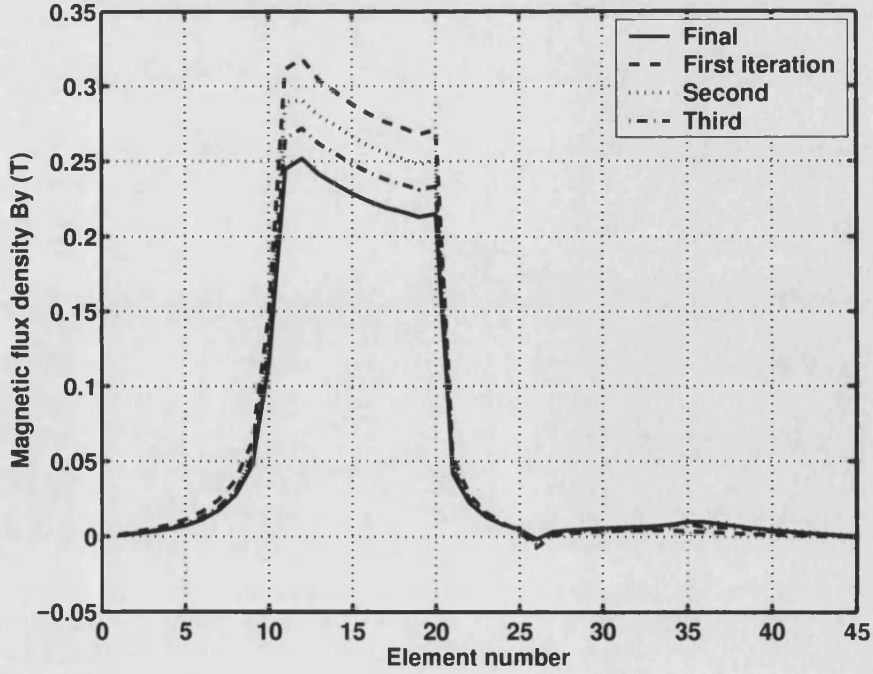


Figure 4.8: B_y for Newton iterations, $Ni = 500\text{A}$, $g = 1mm$, full steps.

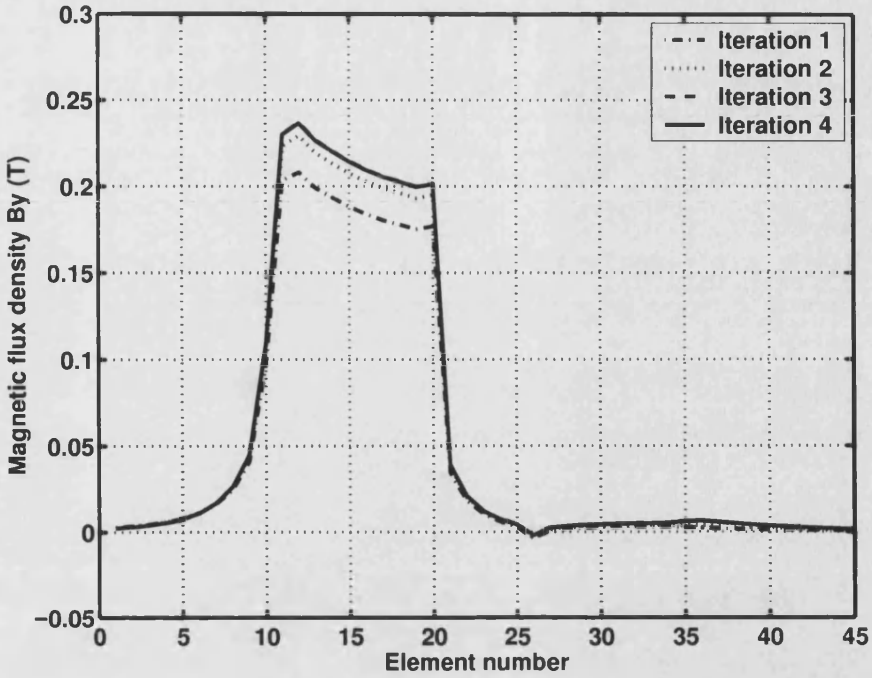


Figure 4.9: B_y for Newton iterations, $Ni = 500\text{A}$, $g = 1mm$, line search steps.

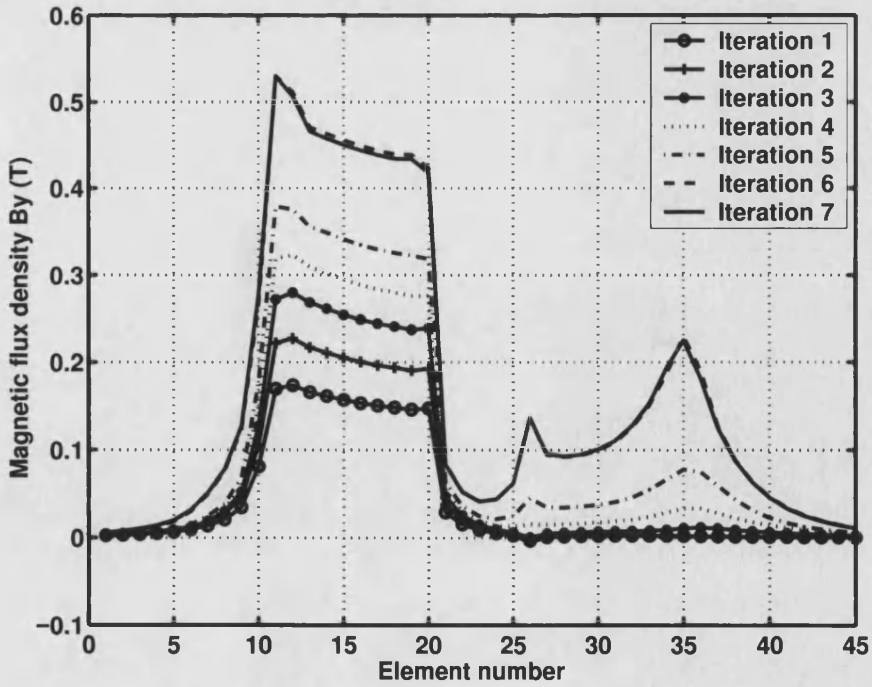


Figure 4.10: B_y for Newton iterations, $Ni = 5000\text{A}$, $g = 1mm$, line search steps.

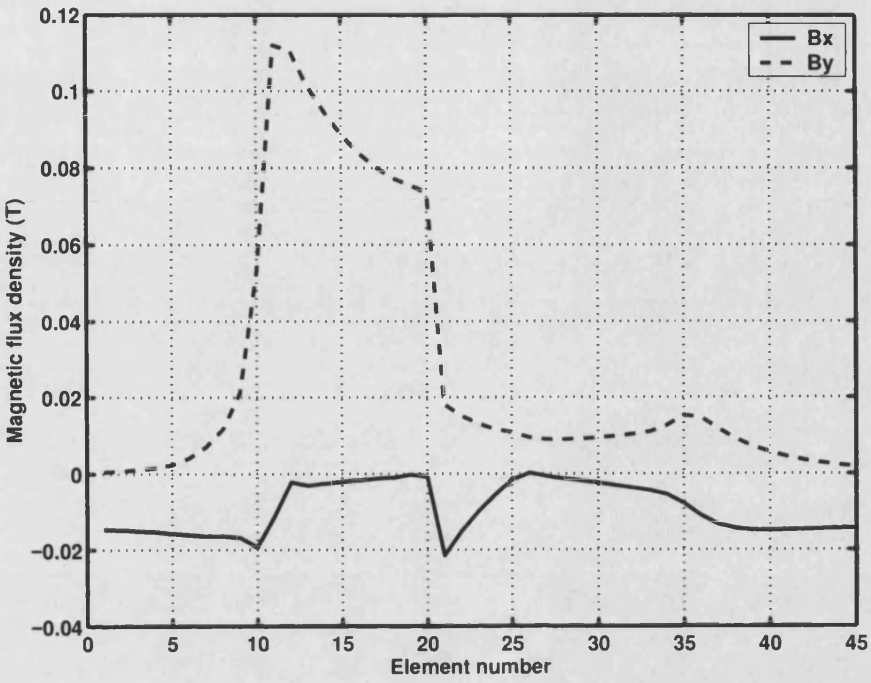


Figure 4.11: Magnetic flux density for Ni =500A.

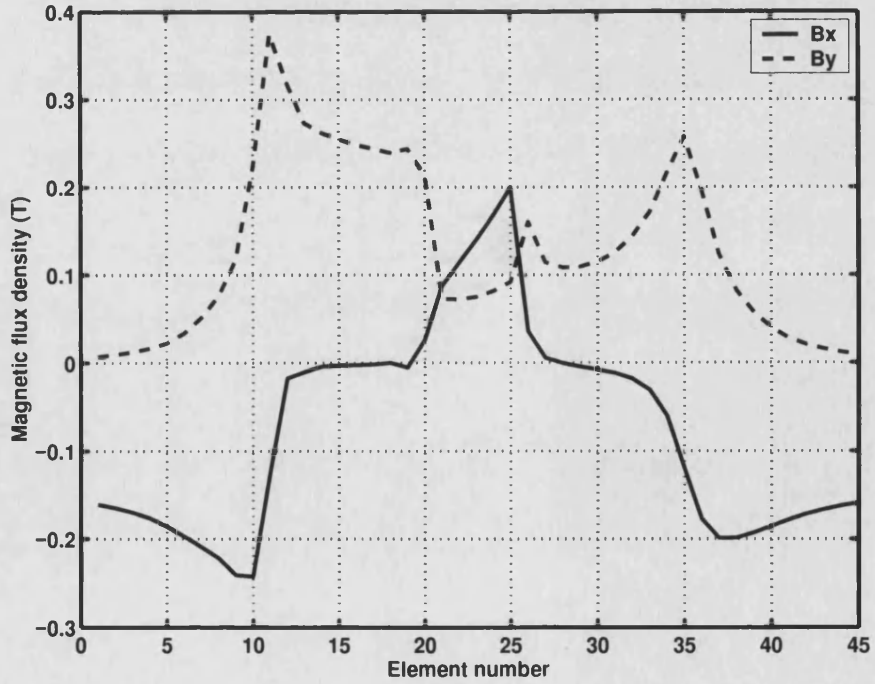


Figure 4.12: Magnetic flux density for Ni =5000A.

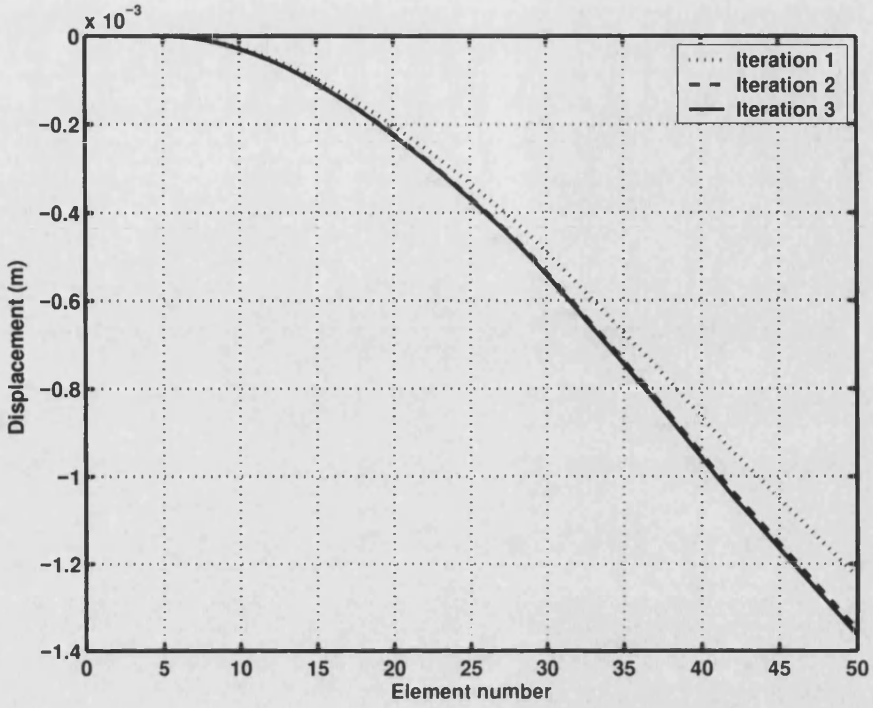


Figure 4.13: Calculated displacement along plate for Gauss–Seidel iteration.

4.4 Solution of the coupled problem

Once the magnetostatics problem has been solved, the elasticity problem can in turn be solved, and the sequence repeated to reach a final state for the coupled problem (Gauss–Seidel iteration). The finite element solution of the elastic problem follows standard procedures [58, 75, 76]; the developed FE elasticity code was tested using published benchmark problems [77]. A 1D beam element formulation is used.

A fast convergence (2 to 3 iterations are enough) is achieved for this problem, as can be seen in Fig.4.13.

Dimensions: 0.234m x 0.04m x 0.0033m
Air gap: 0.013m
Elastic modulus: $100x10^9$ Pa

Table 4.4: Plate characteristics.

4.5 Experimental results

The displacement of a cantilever structure above a magnet was measured and the results compared against those predicted. The dimensions of the magnet and the characteristics of the material appear in [78]. Other dimensions and constants of the problem are summarized in Table4.4. The plate is fixed at one of its ends and only one of the 700-turn coils was connected (the one closer to the plate support).

The elastic modulus of the plate was determined by performing a measurement of the deflection when a known force was applied and using (4.12). The magnetic characteristics of the plate are difficult to obtain because the available tester can not handle samples of so small width (3.3mm); the B-H curve of a material thought to be the same was used instead; this is an untreated mild steel. The curve is shown in Fig.4.14 and its points listed in Table 4.5.

Fig.4.15 shows the measured and calculated displacement at the tip. The calculated displacements are reasonably close to the measurements. Fig.4.16 shows the calculated displacement when an annealed mild steel material is used for the plate, as well as the upper limit (the maximum deflection possible). To obtain this upper limit for the deflection, the problem was solved setting a linear permeability for the plate to a very high value ($\mu_r = 10^6$) to simulate approaching infinity. This value represents the largest possible deflection, achieved by using the best possible material.

H (A/m)	B (T)	H (A/m)	B (T)
0	0	13000	1.815
520	0.22479	13520	1.8265
1040	0.4415	14040	1.8356
1560	0.64774	14560	1.843
2080	0.83985	15080	1.85
2600	1.0054	15600	1.8568
3120	1.1288	16120	1.8636
3640	1.2278	16640	1.8705
4160	1.317	17160	1.8773
4680	1.4021	17680	1.884
5200	1.475	18200	1.8904
5720	1.5313	18720	1.8961
6240	1.5754	19240	1.9015
6760	1.6131	19760	1.9066
7280	1.6413	20280	1.9118
7800	1.6662	20800	1.9171
8320	1.6893	21320	1.9222
8840	1.7104	21840	1.9268
9360	1.724	22360	1.9306
9880	1.737	22880	1.9339
10400	1.75	23400	1.9371
10920	1.763	23920	1.9405
11440	1.776	24440	1.944
11960	1.789	24960	1.9476
12480	1.8021	25480	1.9513

Table 4.5: B-H points, mild steel.

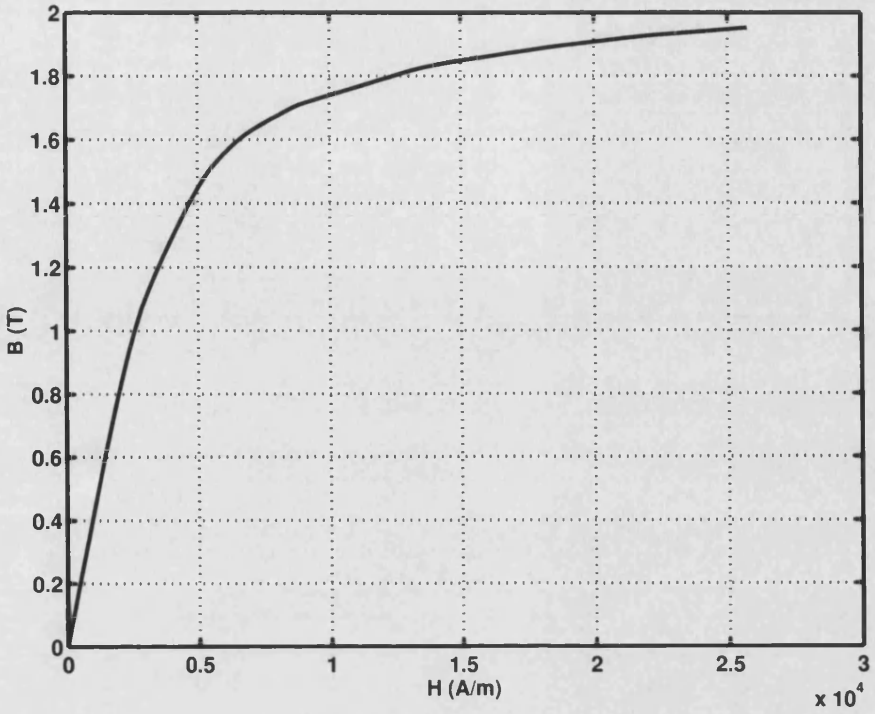


Figure 4.14: BH curve, mild steel.

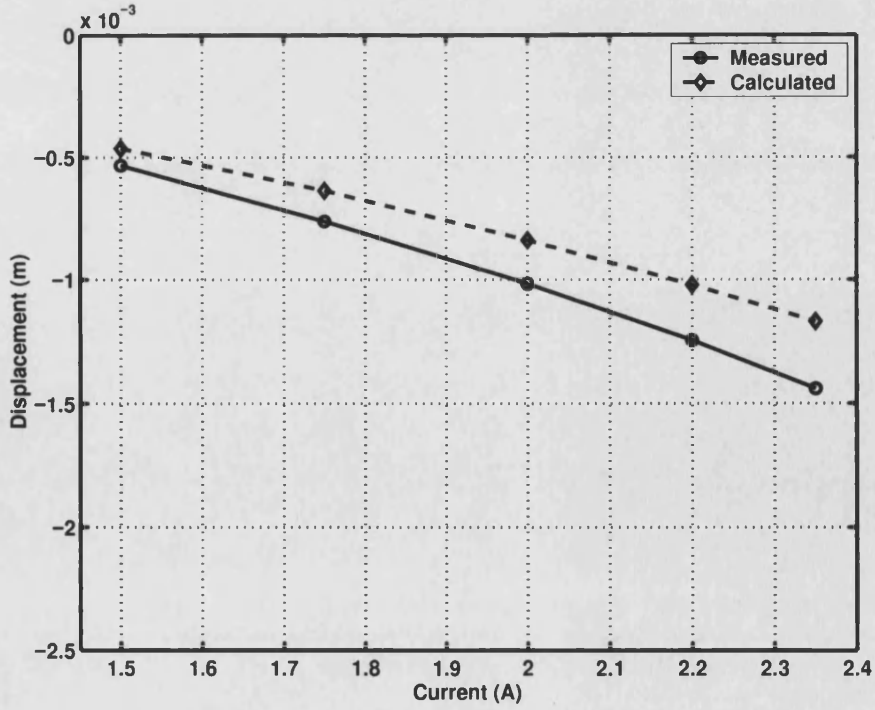


Figure 4.15: Measured and calculated displacements for untreated steel plate.

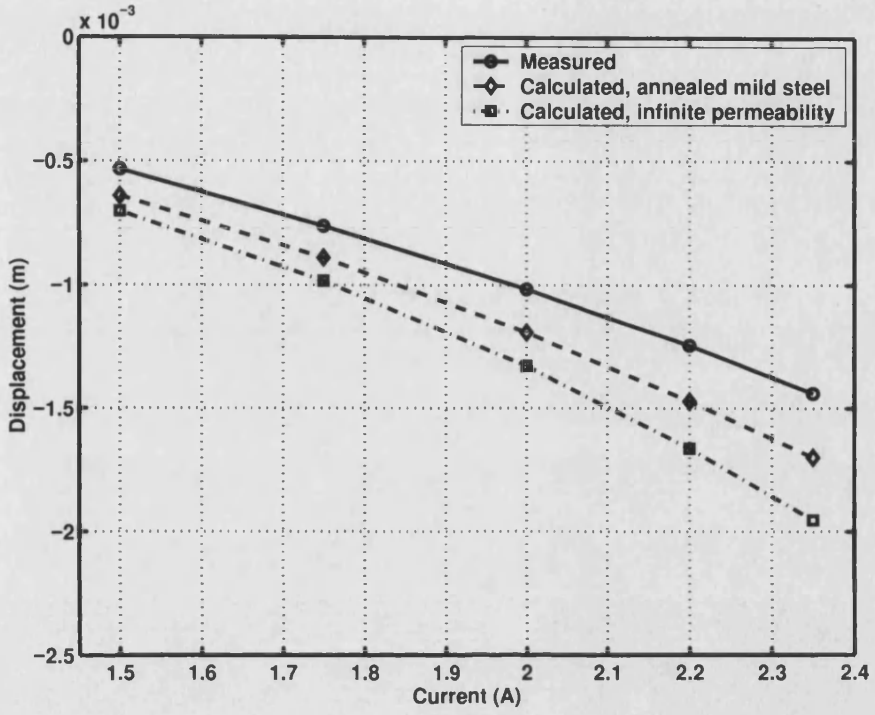


Figure 4.16: Measured and calculated displacements for annealed steel plate and infinite permeability plate.

4.6 Conclusions

The solution of coupled problems involving deformation of the material was discussed. The case of deformation caused by a magnetic force was considered and it was asserted that the deformation may be attributed to the surface force only. The Maxwell stress tensor evaluated at the interface between the ferromagnetic body and the surrounding air is used to determine the magnetic pressure distribution; the coupled problem is the one of fluid-solid interaction.

A problem involving the deflection of a ferromagnetic plate was studied. For the cases considered, the solution of the coupled problem is possible by means of a few Gauss–Seidel iterations. Comparison with experimental results shows encouraging agreement.

The methods discussed can be applied in cases where there is a significant deformation with small stress in the structure, for instance beams and plates and in general slender structures where one of the dimensions is small compared to the others. If this is not the case, then the surface force will not be large enough to cause any noticeable displacement.

Chapter 5

Time Stepping Algorithms for Nonlinear Equations

5.1 Introduction

In this chapter, some algorithms for the integration of systems of ordinary differential equations are investigated, with special reference to nonlinear, first-order initial-value problems resulting from finite element formulations. The system of differential equations of the form

$$\mathbf{C}(\mathbf{x},t)\dot{\mathbf{x}} + \mathbf{K}(\mathbf{x},t)\mathbf{x} + \mathbf{f}(\mathbf{x},t) = \mathbf{0} \tag{5.1}$$

is to be solved over the interval $t_0 \leq t \leq t_f$, given the initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. This is known as an initial value problem.

Equation (5.1) arises, for example, from a finite element discretization of the space variables, applied to a time-dependent problem [62]; it also appears in other instances of discrete analysis, such as electric networks. Various approaches are possible for

solving it, most of them falling in the broad categories of finite difference and finite element discretizations.

The damping matrix $\mathbf{C}$ may be singular, in which case (5.1) represents a differential–algebraic equation (DAE). It is known that methods that work for ordinary differential equations (ODE) do not always work when applied to differential–algebraic equations. Also, the system may be *stiff*, meaning that several modes with widely differing time constants are present. Again, it has been found that methods that work for nonstiff problems fail when applied to stiff ones. These facts have to be taken into account when selecting a time-stepping method.

The evolution of the system is obtained from a recurrence formula in such a fashion that one point at a time is solved for. Because of this, the algorithms are collectively known as time-stepping. Some possibilities are discussed in this chapter, organized as follows: first, the basic finite difference formulas using the forward and backward Euler approximation are presented in a form which makes them applicable to the first order problem (5.1). The resulting expressions have been applied to the coupled electromechanical problems presented in this thesis. A special formula which we call modified Euler method, suitable for solving the rigid body mechanical equation, is briefly discussed. The application of time-stepping schemes to the case of differential-algebraic systems of equations of the form (5.1) is considered next; it is asserted that conventional schemes can be applied in this case too, and modified Adams-Moulton and Gear formulas for that purpose are presented. The implementation of the Adams-Moulton formula is applied to the coupled electromechanical problem and its stability and accuracy characteristics assessed. The popular finite element θ-method is considered next; it is shown that the formula for constant matrices fails when applied to the nonlinear case. Two alternative methods are considered, and we show that they imply a constant damping matrix; this can be a drawback

of the method for certain applications. Finally, the more general BDF (backward differentiation formula) approach is presented and some aspects of its application to finite element problems considered.

The diversity of methods reflects the fact that there is not a best algorithm. It is impossible to single out a method that will outperform all the others in all applications [79].

5.2 Basic finite difference methods

Finite difference approximations of the time derivative result in time stepping formulas. One characteristic of these methods is that the defining equation (5.1) is satisfied only at the point, so that they belong to the point-collocation class of methods. Depending on the finite difference formula used (forward difference or backward difference), the following methods result.

5.2.1 Forward Euler

The forward finite difference formula is

$$\dot{x}^n = \frac{x^{n+1} - x^n}{\Delta t}, \tag{5.2}$$

where $x^n = x(t^n)$ and the time points are marked $t^0, t^1, ..., t^n, t^{n+1}$. This is an explicit formula, because the next state can be found from

$$x^{n+1} = x^n + \Delta t \; \dot{x}^n. \tag{5.3}$$

This means that the next state x^{n+1} can be found once the derivative at a given point $\dot{x}^n$ is known. The formula (5.3) as it stands is valid only for scalar equations. If we attempt to adapt it to the vector equation (5.1), we find that this is sensible only for the global equation. Otherwise, if applied at the element level, we would end with each nodal variable taking a value that is well defined and does not depend on other variables outside the particular element. As a result, distinct values will result for the same nodal variable, predicted by the explicit formula applied to neighbour elements and a coherent global state does not exist.

Thus, if an explicit formula is to be applied, this can be done only at the global level. This implies the evaluation of the global derivative, as required in (5.3). However, we will later see that the evaluation of the derivative is not possible in some cases, for instance for differential-algebraic systems of equations. For these cases, implicit formulas are to be preferred.

The following implicit formula is proposed. Substituting the forward expression for the time derivative (5.2) into (5.1) and requiring that this latter equation be satisfied at $t = t^n$, we arrive to

$$\mathbf{C}^n\mathbf{x}^{n+1} + [\Delta t\ (\mathbf{K}^n\mathbf{x}^n + \mathbf{f}^n) - \mathbf{C}^n\mathbf{x}^n] = \mathbf{0}. \tag{5.4}$$

5.2.2 Backward Euler

The backward finite difference formula is

$$\dot{x}^{n+1} = \frac{x^{n+1} - x^n}{\Delta t}. \tag{5.5}$$

This is an implicit formula, since the expression for the next state contains the derivative at that point:

$$x^{n+1} = x^n + \Delta t\ \dot{x}^{n+1}. \tag{5.6}$$

Substituting the backward expression for the time derivative (5.5) into (5.1) and requiring that this latter equation be satisfied at $t = t^{n+1}$, we arrive to the following

formula:

$$\left(\mathbf{C}^{n+1} + \Delta t\ \mathbf{K}^{n+1}\right)\mathbf{x}^{n+1} + \left(-\mathbf{C}^{n+1}\mathbf{x}^{n} + \Delta t\ \mathbf{f}^{n+1}\right) = \mathbf{0}. \tag{5.7}$$

5.2.3 Modified Euler

For the special case of the scalar second order mechanical equation

$$M\ddot{x} = F_t, \tag{5.8}$$

the following discretization approach has been used by others [80]. Starting from the second order approximation

$$x^{n+1} = x^n + \Delta t\ \dot{x}^n + \frac{\Delta t}{2}\ \ddot{x}^n, \tag{5.9}$$

we see that the next position can be calculated from this explicit expression once we know the velocity $\dot{x}$ and acceleration $\ddot{x}$. The forward finite difference formula may be used to determine the velocity:

$$\dot{x}^n = \dot{x}^{n-1} + \Delta t\ \ddot{x}^n, \tag{5.10}$$

with the acceleration $\ddot{x}^n$ calculated from (5.8) according to

$$\ddot{x}^n = \frac{F_t^n}{M}. \tag{5.11}$$

The resulting scheme is explicit and assumes that the external forces which have been grouped into F_t are not functions of the velocity.

5.3 Differential–algebraic equations

The differential equation

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, t), \tag{5.12}$$

has been the subject of considerable attention. Several algorithms are available for its solution, for instance the Runge-Kutta and Adams schemes [81]. These may be considered as finite difference methods because they attempt to satisfy (5.12) at a discrete number of points.

An equation of the form (5.1) has some resemblance with (5.12). It would appear that these two equations can be made equivalent simply by re-writing (5.1) in the form

$$\dot{\mathbf{x}} = \mathbf{C}^{-1}\left[-\mathbf{K}\mathbf{x} - \mathbf{f}\right], \tag{5.13}$$

but this requires that $\mathbf{C}$ be nonsingular. This requirement is very restrictive; for instance, in the case of the finite element formulation of the eddy current problem, the global $\mathbf{C}$ matrix is singular [61]. In this case, the singularity arises because some rows of the damping matrix $\mathbf{C}$ in (5.1) are zero. The system (5.1) constitutes a special case of a differential–algebraic system of equations, that is a system comprising both differential and algebraic equations.

We have seen in a previous section that specially derived expressions can handle directly equations in the form (5.1), for instance the Backward Euler scheme. It would be very convenient if we can as well use some of the available formulas for equations

of the form (5.12). The singularity of the global damping matrix $\mathbf{C}$ prevents this; on the other hand, the time discretization formula may be applied at the element level with the added advantage that the assembly and solution process are simplified (since they are the same as for a static case).

The element equation is of the same form as the global equation:

$$\mathbf{C}_e(\mathbf{x}_e, t)\dot{\mathbf{x}}_e + \mathbf{K}_e(\mathbf{x}_e, t)\mathbf{x}_e + \mathbf{f}_e(t) = \mathbf{0}. \tag{5.14}$$

The singularity in the global matrix $\mathbf{C}$ arises from the fact that the local matrix $\mathbf{C}_e$ vanishes for some elements. In such case, we would simply assemble the expression

$$\mathbf{K}_e(\mathbf{x}_e, t)\mathbf{x}_e + \mathbf{f}_e(t) = \mathbf{0} \tag{5.15}$$

in the usual way. No time stepping algorithm is necessary for those elements since their evolution is governed by the simpler relation (5.15), which does not include derivatives. For those elements where the $\mathbf{C}_e$ matrix exists, equation (5.14) must be assembled. This can not be done directly because of the presence of the derivative term; instead, some form of discretization of the time must be applied first. Here is where the finite difference formulas come into play. The process can be illustrated by considering the backward Euler scheme applied at the element level:

$$\left(\mathbf{C}_e^{n+1} + \Delta t\ \mathbf{K}_e^{n+1}\right)\mathbf{x}^{n+1} + \left(-\mathbf{C}^{n+1}\mathbf{x}^n + \Delta t\ \mathbf{f}^{n+1}\right) = \mathbf{0}. \tag{5.16}$$

Given an initial state $\mathbf{x}_n$, a time stepping scheme based on formula (5.16) is:

$$\begin{cases} \mathbf{K}_e^{n+1}\mathbf{x}_e^{n+1} + \mathbf{f}_e^{n+1} &= \mathbf{0}, \\ \left(\mathbf{C}_e^{n+1} + \Delta t\mathbf{K}_E^{n+1}\right)\mathbf{x}_e^{n+1} + \left(-\mathbf{C}_e^{n+1}\mathbf{x}_e^n + \Delta t\ \mathbf{f}_e^{n+1}\right) &= \mathbf{0}. \end{cases} \tag{5.17}$$

This scheme has the peculiarity that the assembled equation is dependent on the type of element, i.e. the first expression is applied if $\mathbf{C}_E$ vanishes, and the second otherwise. We expect the properties of the conventional backward Euler method apply in this case too, namely first order accuracy and unconditional stability.

5.4 Other finite difference methods

The ideas of the previous section open the possibility of using known formulas originally developed for differential equations of the form (5.12) to the differential-algebraic case (5.1). We can distinguish two families of formulas: one based on a Taylor series expansion (Runge–Kutta or single step methods) and the other based on interpolation (Adams or multistep methods).

5.4.1 Runge–Kutta methods

Starting from equation (5.12), a Taylor series expansion can be used to develop formulas of increasing order for $\mathbf{x}^{n+1}$. However, additional derivatives must be evaluated, which is hard to implement in practice. The Runge-Kutta methods remedy this drawback by searching an expression containing no derivatives other than the first (which is known or is implicit), while still retaining a given accuracy [81, 82]. Explicit Runge–Kutta formulas are not useful in our case, as explained above, while implicit Runge–Kutta have been used for the global equation, see for instance [83]).

5.4.2 Adams methods

Adams–Bashforth algorithms, being explicit, are not very useful in the present case as explained above. Adams–Moulton formulas are implicit, so they are well suited to our purposes. The formulas are multistep, meaning that they make use of the solution in preceding time steps; this also implies that the methods of third order onwards are not self-starting. Some of the formulas adapted to our case are:

First order Adams–Moulton (backward Euler) algorithm:

$$(\mathbf{C}_{n+1} + \Delta t \mathbf{K}_{n+1})\,\mathbf{x}_{n+1} + (-\mathbf{C}_{n+1}\mathbf{x}_n + \Delta t \mathbf{f}_{n+1}) = \mathbf{0}. \tag{5.18}$$

Second order Adams–Moulton (trapezoidal) algorithm:

$$\left(\mathbf{C}_{n+1} + \frac{\Delta t}{2}\mathbf{K}_{n+1}\right)\mathbf{x}_{n+1} + \left[-\mathbf{C}_{n+1}\left(\mathbf{x}_n + \frac{\Delta t}{2}\dot{\mathbf{x}}_n\right) + \frac{\Delta t}{2}\mathbf{f}_{n+1}\right] = \mathbf{0}. \tag{5.19}$$

Third order Adams–Moulton algorithm:

$$\left(\mathbf{C}_{n+1} + \frac{5\Delta t}{12}\mathbf{K}_{n+1}\right)\mathbf{x}_{n+1}\left[-\mathbf{C}_{n+1}\left(\mathbf{x}_n + \frac{8\Delta t}{12}\dot{\mathbf{x}}_n - \frac{\Delta t}{12}\dot{\mathbf{x}}_{n-1}\right) + \frac{5\Delta t}{12}\mathbf{f}_{n+1}\right] = \mathbf{0}. \tag{5.20}$$

Formulas for orders 4, 5 and 6 can be developed. The first and second order algorithms (5.18, 5.19) are A-stable, but Adams–Moulton algorithms of order 3 or more do not have that characteristic [82]. A-stability is an important feature if the method is to be used to solve stiff equations, e.g. eddy current problems [84].

The presented formulas require the knowledge of past derivatives, which results in expensive code because additional storage is required. For problems with small numbers of variables, the scheme is worth investigating; for that purpose we use a lumped parameter model of an induction motor. The equations, put in the form of (5.1) are

$$\begin{bmatrix} \mathbf{L}(\theta) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J(t) & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{I}} \\ \dot{\omega} \\ \dot{\theta} \end{bmatrix} +$$

$$\begin{bmatrix} \mathbf{R}+\omega\frac{\partial \mathbf{L}}{\partial\theta} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B(\omega,t)+\frac{\partial J}{\partial t} & 0 \\ \mathbf{0} & -1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \omega \\ \theta \end{bmatrix} + \begin{bmatrix} -\mathbf{V} \\ T_l(t)-\frac{1}{2}\mathbf{I}^T\frac{\partial \mathbf{L}}{\partial\theta}\mathbf{I} \\ 0 \end{bmatrix} = \mathbf{0}. \qquad (5.21)$$

This problem is small and has a non-singular damping matrix; it is possible therefore to transform it into the explicit form (5.12) at each time step and use some of the available ODE solvers to have a reference against which to compare our results. We use the Matlab solver ODE113, which is a code based on what its authors call numerical differentiation formulas, a variation of the backward differentiation formulas discussed later, [85]. Figures 5.1 and 5.2 show the simulated current and angular velocity, respectively; the number of time steps for each solver is shown in the legends in the figures. ODE113 adapts the step size; we compare it against the second order Adams–Moulton formula (5.19) with fixed step size. It can be appreciated that the results are quite good in this case: both accuracy and stability are satisfactory.

However, for a different mechanical load, the results are not so good. In Fig. 5.3 we appreciate a sizable error in the angular velocity, even with small time steps. This could well be attributed to the lack of step size adaptation, but experience with other problems has been that the Adams–Moulton formulas do not provide much more improved accuracy compared to simpler, first order formulas like the Backward Euler scheme. Moreover, the Adams–Moulton formulas are less stable than the Backward Euler scheme. This is demonstrated by using the rotational test rig; the results of the simulation are plotted in Figures 5.4 to 5.6. With a time step size of 1ms, the predicted position is reasonably close to the measured values as can be appreciated in Fig. 5.4; however, for a time step of 5ms the difference is sizable, Fig. 5.5; more importantly, the first order formula performs better in this case. Finally, for a time step of 25ms, the stability is compromised and the results rendered useless, Fig. 5.6.

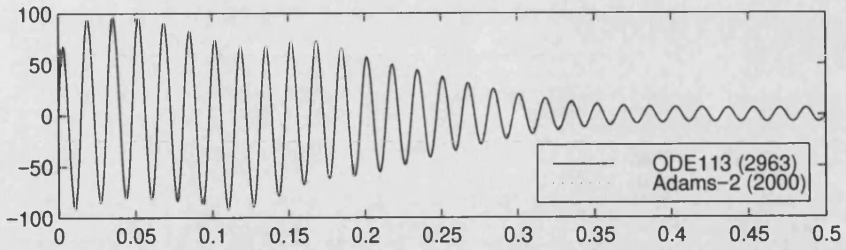


Figure 5.1: Induction motor, phase a stator current (A) vs. Time (s).

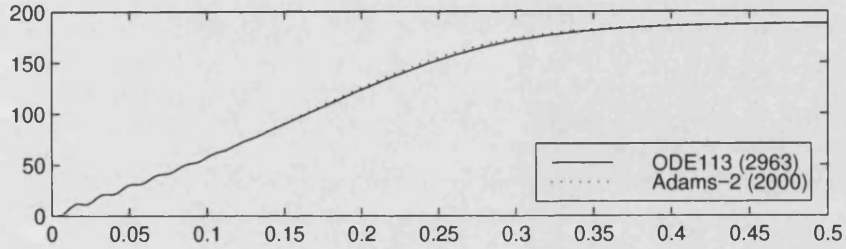


Figure 5.2: Induction motor, angular velocity (rad/s) vs. Time (s).

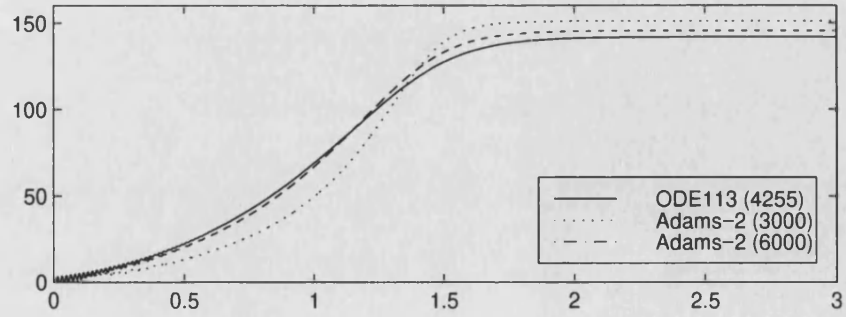


Figure 5.3: Induction motor with large inertia, angular velocity (rad/s) vs. Time (s).

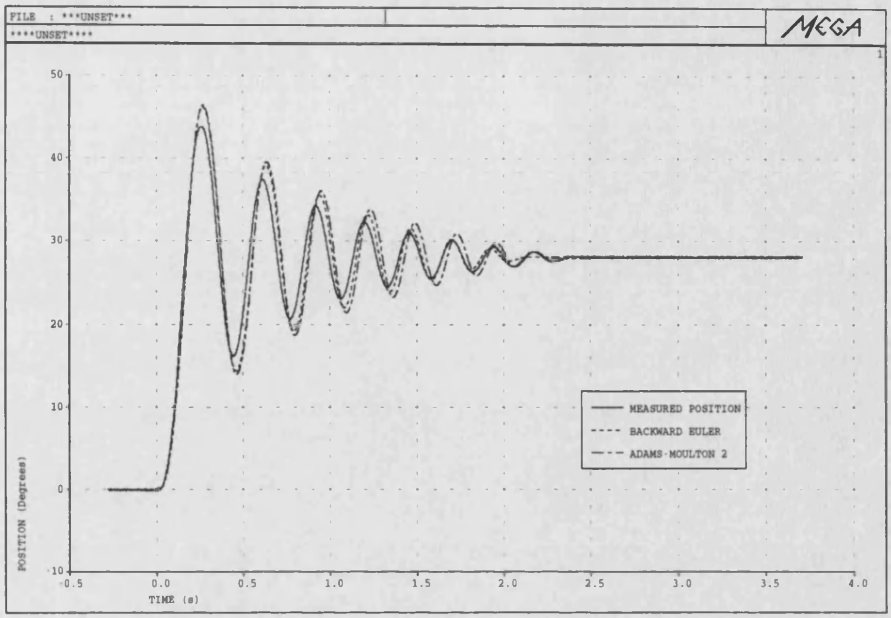


Figure 5.4: Rotational test rig position, time step 1ms.

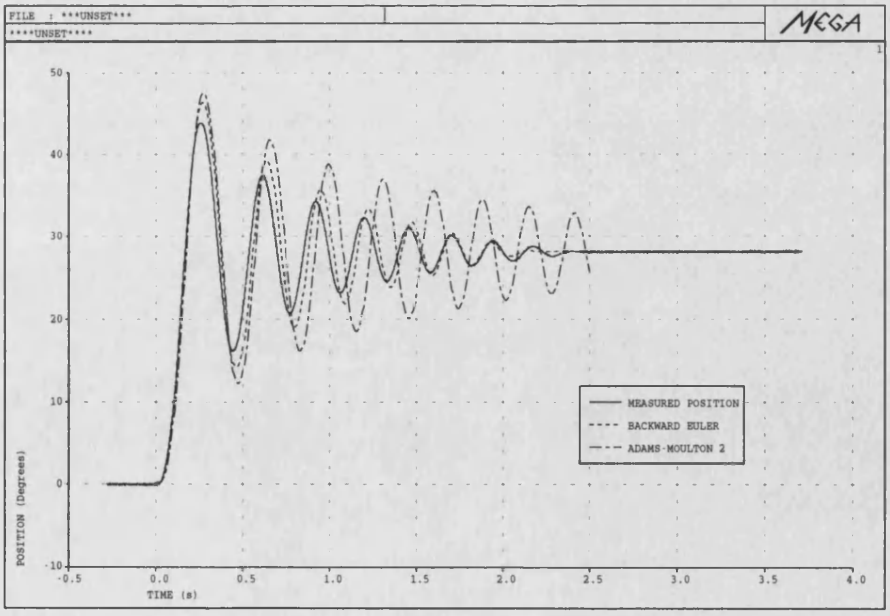


Figure 5.5: Rotational test rig position, time step 5ms.

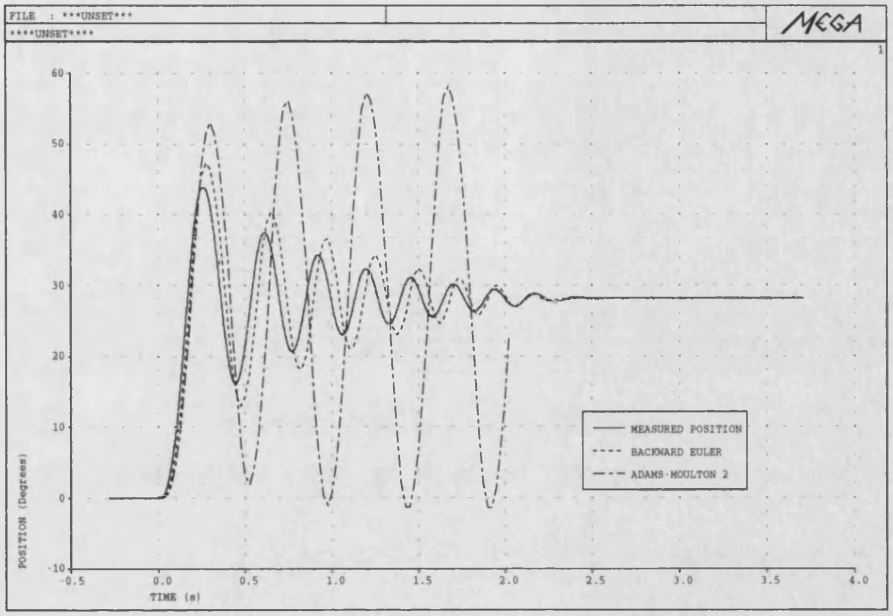


Figure 5.6: Rotational test rig position, time step 25ms.

5.4.3 Gear's method

A family of formulas aimed at stiff equations was derived by Gear [82]. Specifically, higher order formulas are A-stable:

Gear's second order algorithm:

$$\left(\mathbf{C}_{n+1} + \frac{2\Delta t}{3}\mathbf{K}_{n+1}\right)\mathbf{x}_{n+1} + \left[\mathbf{C}_{n+1}\left(-\frac{4}{3}\mathbf{x}_n + \frac{1}{3}\mathbf{x}_{n-1}\right) + \frac{2\Delta t}{3}\mathbf{f}_{n+1}\right] = \mathbf{0}. \quad (5.22)$$

Gear's third order algorithm:

$$\left(\mathbf{C}_{n+1} + \frac{6\Delta t}{11}\mathbf{K}_{n+1}\right)\mathbf{x}_{n+1} +$$

$$\left[\mathbf{C}_{n+1}\left(-\frac{18}{11}\mathbf{x}_n + \frac{9}{11}\mathbf{x}_{n-1} - \frac{2}{11}\mathbf{x}_{n-2}\right) + \frac{6\Delta t}{11}\mathbf{f}_{n+1}\right] = \mathbf{0}. \quad (5.23)$$

Formulas for orders 4, 5 and 6 can be developed. The size of the time step is thus only limited by the allowable error. If an efficient solution is to be obtained, a variable step size must be used. This requirement arises from the nature of stiff systems: a rapidly vanishing response requires a small step size, and a large step size is desirable when only the slow varying response remains.

Both Adams–Moulton and Gear's formulas use a fixed time step size. Thus, changing this size requires that a new set of past values must be interpolated for the transition to be achieved. The interpolation using directly the $\mathbf{x}$ values has a considerable computational cost, so that special storage schemes have been developed to alleviate the burden; these are the backward difference vector and the Nordsieck vector [81, 82]. However, besides requiring a new storage scheme, a set of linear algebraic equations must be solved for *each* dynamic variable x_i (in the case of the Nordsieck representation, the matrix of the system of equations is diagonal).

The formulas presented here resemble the more general BDF approach discussed later. The implementation follows the same lines and the only changes are the coefficients in the formulas. See the relevant section for application details.

5.5 Finite element methods

If the finite element approximation process is applied to the time variable, alternative schemes may be devised. Implicit in the finite element idea is the integration over a domain, a time interval in this case. This is a fundamental difference of this family of schemes with respect to the algorithms considered in previous sections, in which the equations are satisfied at single, isolated points (hence their classification as finite difference methods). There is one point of contact between the two approaches, however: if the weighting function is an impulse located at the end of the interval, the equation is satisfied at a discrete point rather than at an interval. Such an integration is called point-collocation.

A frequently used finite element time stepping process is the θ-method, considered next.

5.5.1 Linear invariant systems

The time dependence of $\mathbf{x}$ is represented by the linear approximation

$$\mathbf{x}(t) \approx \mathbf{x}_n \left(1 - \frac{\tau}{\Delta t}\right) + \mathbf{x}_{n+1} \left(\frac{\tau}{\Delta t}\right) = \sum_{i=n}^{n+1} \mathbf{x}_i N_i, \tag{5.24}$$

with $\tau = t - t_n$ and $\Delta t = t_{n+1} - t_n$. Note that the shape functions N are linear (first order). A recurrence scheme is obtained by requiring a null weighted residual:

$$\int_0^{\Delta t} w \left(\mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} + \mathbf{f}\right) d\tau = \mathbf{0}. \tag{5.25}$$

Substituting (5.24) into (5.25):

$$\int_0^{\Delta t} w \left[\mathbf{C}\left(\mathbf{x}_n \dot{N}_n + \mathbf{x}_{n+1}\dot{N}_{n+1}\right) + \mathbf{K}\left(\mathbf{x}_n N_n + \mathbf{x}_{n+1} N_{n+1}\right) + \mathbf{f}\right] d\tau = \mathbf{0}, \qquad (5.26)$$

Grouping:

$$\left(\int_0^{\Delta t} \frac{w}{\Delta t}\mathbf{C} d\tau + \int_0^{\Delta t} \frac{w}{\Delta t}\mathbf{K}\tau d\tau\right)\mathbf{x}_{n+1}$$

$$+\left[-\int_0^{\Delta t} \frac{w}{\Delta t}\mathbf{C} d\tau + \int_0^{\Delta t} w\mathbf{K}(1-\frac{\tau}{\Delta t}) d\tau\right]\mathbf{x}_n + \int_0^{\Delta t} w\mathbf{f} d\tau = \mathbf{0}. \qquad (5.27)$$

In the special case of constant matrices $\mathbf{C}$ and $\mathbf{K}$, these can be taken out of the integral sign, and letting

$$\theta = \left(\frac{1}{\Delta t}\right)\frac{\int_0^{\Delta t} w\tau d\tau}{\int_0^{\Delta t} w d\tau}, \qquad (5.28)$$

(5.27) can be put in the simpler form:

$$\left(\frac{\mathbf{C}}{\Delta t} + \theta\mathbf{K}\right)\mathbf{x}_{n+1} + \left[-\frac{\mathbf{C}}{\Delta t} + (1-\theta)\mathbf{K}\right]\mathbf{x}_n + \frac{\int_0^{\Delta t} w\mathbf{f} d\tau}{\int_0^{\Delta t} w d\tau} = \mathbf{0}. \qquad (5.29)$$

The next state $\mathbf{x}_{n+1}$ is obtained by solving the linear equation (5.29). This time-marching method is unconditionally stable for $\theta \geq 1/2$ [62].

In order to test the time stepping scheme above, we use the induction motor equation (5.21). We consider several special cases of this equation, resulting in transient problems with different characteristics, and apply the θ-method given by (5.29). Again, the results are compared against those given by the ODE113 solver [85].The first case is the induction motor with locked rotor; the equations will become linear

and with constant coefficients:

$$\mathbf{L}\dot{\mathbf{I}} + \mathbf{R}\mathbf{I} - \mathbf{V} = \mathbf{0}. \tag{5.30}$$

The resulting current is shown in Fig.5.7. We conclude that the scheme is up to the task.

A second case arises when we allow a varying moment of inertia J and model the electrical part as an equivalent circuit. This results in a reduced set of linear equations with variable coefficients:

$$\begin{bmatrix} J(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} B(\omega,t) + \frac{\partial J}{\partial t} & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ \theta \end{bmatrix} + \begin{bmatrix} T_l(t) - T_e(\omega) \\ 0 \end{bmatrix} = \mathbf{0}. \tag{5.31}$$

The results for this case are shown in Fig.5.8. A good correspondence is attained, but the accuracy is compromised. A time step adaptation scheme is desirable in this case and consequently a very basic algorithm for step adaptation was implemented; the results for the test case are shown in Fig.5.9. We conclude that the θ-method is able to give acceptable answers for systems with constant coefficients and even for systems with varying coefficients; the addition of time adaptation makes it very competitive.

Lastly, the full nonlinear equation (5.21) is solved. The results are shown in Figs.5.10 and 5.11, from where it is evident that the scheme loses stability. This is caused by the nature of the system being integrated and is not surprising since the system coefficients were assumed constant in the derivation of the formula.

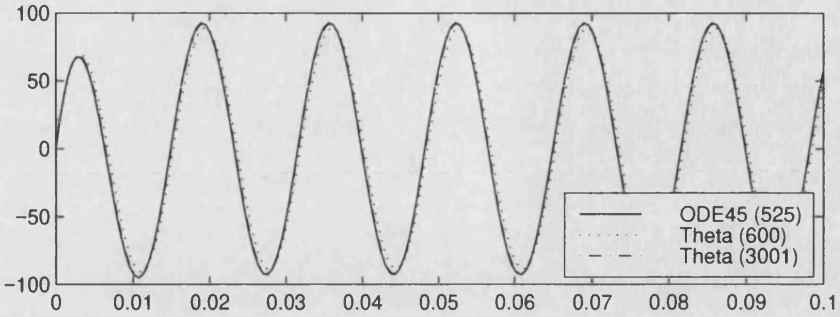


Figure 5.7: Induction motor with locked rotor, phase a stator current (A) vs. Time (s).

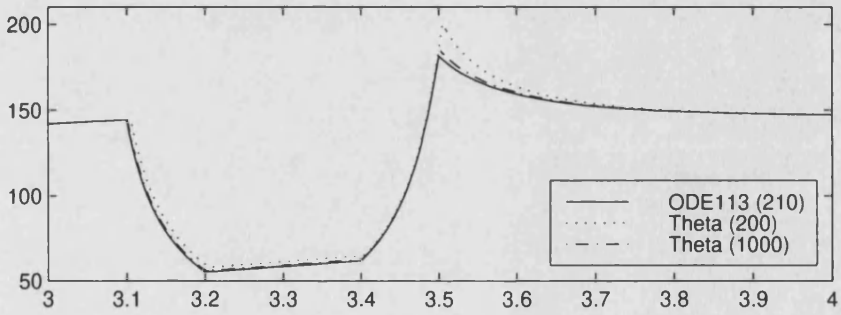


Figure 5.8: Induction motor with variable inertia, angular velocity (rad/s) vs. Time (s).

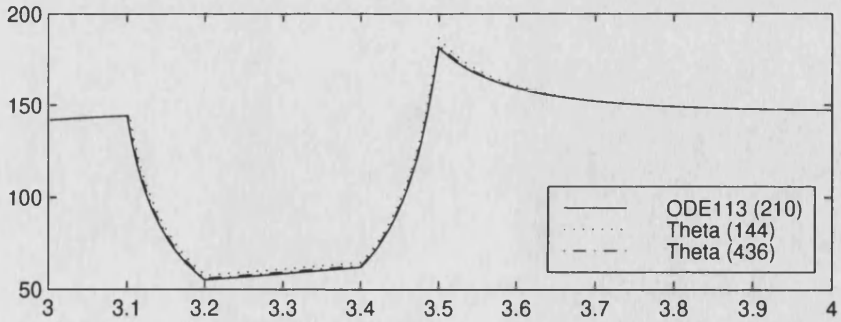


Figure 5.9: Induction motor with variable inertia, angular velocity (rad/s) vs. Time (s). Case with step adaptation.

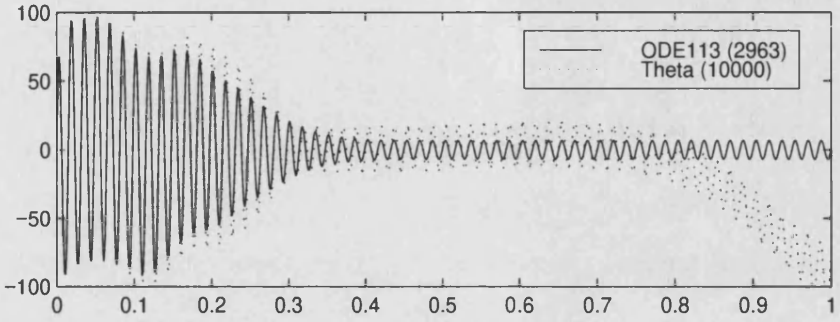


Figure 5.10: Induction motor, nonlinear equation. Stator current (A) vs. Time (s).

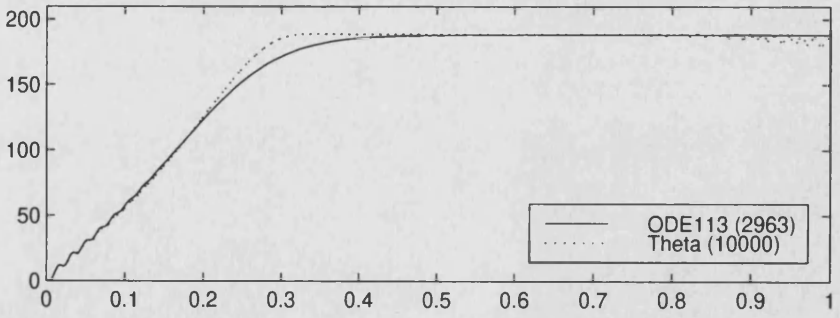


Figure 5.11: Induction motor, nonlinear equation. Angular velocity (rad/s) vs. Time (s).

5.5.2 Nonlinear systems

As it stands, the scheme (5.29) can only be applied to linear problems with constant matrices. In order to adapt it to the nonlinear case, the following matrix-vector products are defined:

$$\mathbf{p} = \mathbf{C}\mathbf{x}, \tag{5.32}$$

$$\mathbf{q} = \mathbf{K}\mathbf{x}. \tag{5.33}$$

Starting with (5.26) and following the same procedure as above, we arrive to

$$\left(\frac{\mathbf{p}_{n+1}}{\Delta t} + \theta \mathbf{q}_{n+1}\right) + \left[-\frac{\mathbf{p}_n}{\Delta t} + (1-\theta)\mathbf{q}_n + \frac{\int_0^{\Delta t} w\mathbf{f}\, d\tau}{\int_0^{\Delta t} w\, d\tau}\right] = \mathbf{0}, \tag{5.34}$$

Using the definitions (5.32) and (5.33) the following nonlinear algebraic equation is obtained [86]:

$$\left(\frac{\mathbf{C}_{n+1}\mathbf{x}_{n+1}}{\Delta t} + \theta \mathbf{K}_{n+1}\mathbf{x}_{n+1}\right) + \left[-\frac{\mathbf{C}_n\mathbf{x}_n}{\Delta t} + (1-\theta)\mathbf{K}_n\mathbf{x}_n + \frac{\int_0^{\Delta t} w\mathbf{f}\, d\tau}{\int_0^{\Delta t} w\, d\tau}\right] = \mathbf{0}. \tag{5.35}$$

The solution of this nonlinear equation is required at each time step. The calculation of the Jacobian follows the procedure described in a later section.

Implicit in the derivation of (5.35) are the approximations

$$\mathbf{C}\dot{\mathbf{x}} \approx \mathbf{C}_n\mathbf{x}_n\dot{N}_n + \mathbf{C}_{n+1}\mathbf{x}_{n+1}\dot{N}_{n+1}, \tag{5.36}$$

$$\mathbf{Kx} \approx \mathbf{K}_n \mathbf{x}_n N_n + \mathbf{K}_{n+1} \mathbf{x}_{n+1} N_{n+1}. \tag{5.37}$$

When the expressions for the derivatives of the shape functions $\dot{N}_n$ and $\dot{N}_{n+1}$ are replaced, we see that the approximation (5.36) is really

$$\mathbf{C}\dot{\mathbf{x}} \approx \frac{\mathbf{C}_{n+1}\mathbf{x}_{n+1} - \mathbf{C}_n \mathbf{x}_n}{\Delta t} \approx \frac{d}{dt}\mathbf{Cx}, \tag{5.38}$$

which is correct only for a constant matrix $\mathbf{C}$. On the other hand, the expression (5.37) represents the product $\mathbf{K}(\mathbf{x})\mathbf{x}$ as a linear, smoothly varying function. The origin of these difficulties lies in the attempt to use an integrated form, as required by the finite element methodology, when the integrand can not be approximated in a consistent way. For this reason, point collocation methods for nonlinear problems seem to be more suitable. Such point-collocation scheme is achieved in (5.35) by making $\theta = 1$ [86]. The result is the familiar Backward Euler scheme (5.7).

5.5.3 A time-stepping scheme with linearization

We have assumed that when a nonlinear time-transient problem is to be solved, the time stepping algorithm is applied to the system equations (5.1) first, so that at each time step we require the solution of a nonlinear algebraic equation of the form

$$\mathbf{A}^{n+1}\mathbf{x}^{n+1} + \mathbf{b}^{n+1} = \mathbf{0}. \tag{5.39}$$

If the Newton scheme is used, what we are using is in fact a linearized form of (5.39) at each iteration. This order may be inverted partially; the idea is to linearize the $\mathbf{Kx}$ term ($\mathbf{q}$ in (5.33)) of the system equation (5.1) before applying the time

discretization formula. The linearization uses a truncated Taylor series expansion:

$$\mathbf{K}(\mathbf{x})\mathbf{x} = \mathbf{q}(\mathbf{x}) \approx \mathbf{q}_n + \frac{\partial \mathbf{q}}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_n), \tag{5.40}$$

where $\frac{\partial \mathbf{q}}{\partial \mathbf{x}} = \mathbf{J}_\mathbf{q}$ is the Jacobian matrix for $\mathbf{q}$. The difficulties arising from the time variation of this product, mentioned in the previous section, are now avoided by the linearization.

Making use of (5.24) and following the same procedure as before, we obtain

$$(\mathbf{C} + \theta \mathbf{J}_{\mathbf{q},n})\,\mathbf{x}_{n+1} + [-\mathbf{C} - \mathbf{J}_{\mathbf{q},n}\,(2\Delta t - \theta)]\,\mathbf{x}_n + \Delta t \left(\mathbf{q}_n + \frac{\int_0^{\Delta t} w f d\tau}{\int_0^{\Delta t} w d\tau}\right) = \mathbf{0}. \tag{5.41}$$

This assumes that $\mathbf{C}$ is linear (constant), otherwise the integration is not possible.

5.6 Backward differentiation formulas

The preceding sections have presented several methods for solving the implicit equation (5.1). Specially developed finite difference formulas were presented, as well as adaptations of other schemes originally developed to handle only explicit equations like (5.12). The finite element approach has been reviewed and some known formulas considered; it was found that the consistent integration of the nonlinear quantities involved constitutes a difficulty. Often, the point collocation equivalent of the θ-method is used, but this amounts to using a finite difference formula in the first place.

Another aspect to be remembered is the possibility of using step adaptation. Although step adaptation of the θ-method is theoretically possible, there exists a much more extensive knowledge of step adaptation for finite difference formulas. In this section we consider yet another method, which appears to address the concerns just expressed. This very general method using the backward differentiation formulas (BDF) can be used to handle (5.1) directly. It offers the additional advantages of easy implementation of variable order and variable step size, and is considered a good choice for stiff problems [87]. We will consider the BDF method in some detail, using finite element terminology.

The general development is due to Gear [88]. Here, we will be discussing the Brayton–Gustavson–Hachtel BDF algorithm [89]. Like the Gear algorithm of a previous section, the BDF is well suited to stiff equations and besides it is more stable for large step size variations [89]. It has no restrictions over the singularity of the $\mathbf{C}$ matrix, so that it can be applied both at the local and at the global level.

The equation to solve is now the general implicit differential–algebraic system

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t) = \mathbf{0}, \tag{5.42}$$

of which the original equation (5.1) is a special case. Suppose the solution of (5.42) in the previous time points $t_n, t_{n-1}, \ldots, t_{n+1-k}$ is known to be $x_n, x_{n-1}, \ldots, x_{n+1-k}$ (we use a scalar variable x here; the extension to the vectorial case is direct). The derivative term is replaced by the discrete formula

$$\dot{x}_{n+1} = -\frac{1}{h}\sum_{i=0}^{k} \alpha_i x_{n+1-i} \tag{5.43}$$

which depends on past values of x and the unknown x_{n+1}; (5.43) receives the name of backward differentiation formula (BDF). We can now seek a point-collocation solution at $t = t_{n+1}$ by inserting (5.43) into (5.42) and solving the nonlinear set of equations for x_{n+1}.

5.6.1 Derivation of the α coefficients

The α coefficients are chosen so as to make formula (5.43) exact whenever $x(t)$ is a polynomial of order k or less. To derive an expression for them there are several possibilities. One is to use:

$$x(t) = N_{n+1}x_{n+1} + N_n x_n + \ldots + N_{n+1-k}x_{n+1-k} = \sum_{i=0}^{k} N_{n+1-i}x_{n+1-i}, \tag{5.44}$$

where the shape functions $N(t)$ are polynomials of order k. Differentiating we get

$$\dot{x}(t) = \sum_{i=0}^{k} \dot{N}_{n+1-i}x_{n+1-i}, \tag{5.45}$$

which is the required formula. To determine the shape functions N and their derivatives $\dot{N}$, some well known procedures might be applied. One possibility is to write for instance:

$$N_1 = p_0 + p_1 t + p_2 t^2 + \ldots + p_k t^k \tag{5.46}$$

and calculate the coefficients p_i by solving the linear equation

$$\begin{bmatrix} 1 & t_{n+1} & t_{n+1}^2 & \cdots & t_{n+1}^k \\ 1 & t_n & t_n^2 & \cdots & t_n^k \\ 1 & t_{n-1} & t_{n-1}^2 & \cdots & t_{n-1}^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_{n+1-k} & t_{n+1-k}^2 & \cdots & t_{n+1-k}^k \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_k = \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{5.47}$$

A similar linear equation must be written and solved for each of the $k+1$ shape functions N_i. In practice, $k \leq 5$ and consequently six 6x6 linear equations must be solved at each time step, or the factors of the square matrix in (5.47) obtained and six backward-forward substitutions performed.

Another approach is to write a Lagrange polynomial for each N_i, f.i. for $k = 3$, $n = 2$, $i = 3$:

$$N_0 = \frac{(t - t_1)(t - t_2)(t - t_3)}{(t_0 - t_1)(t_0 - t_2)(t_0 - t_3)} \tag{5.48}$$

and calculate the derivative directly:

$$\dot{N}_0 = \frac{(t - t_2)(t - t_3) + (t - t_1)(t - t_3) + (t - t_1)(t - t_2)}{(t_0 - t_1)(t_0 - t_2)(t_0 - t_3)}. \tag{5.49}$$

For $k \leq 5$, this implies having $6+5+\ldots+1 = 21$ functions returning the appropriate coefficients, or a general function evaluating them according to some argument.

The two methods just presented are cumbersome to apply; besides, the derivative approximation (5.45) is needed only at the point t_{n+1}, not in the whole interval $[t_{n+1-k}, t_{n+1}]$ since we seek a point collocation solution there, as opposed to a full domain integration. A more efficient method of finding the α coefficients can be constructed by writing

$$x(t) = q_0 + \sum_{i=1}^{k} q_i \left(\frac{t_{n+1} - t}{h} \right)^i . \tag{5.50}$$

This expression can be regarded as a hierarchical shape function, while (5.44) uses standard shape functions [90]. Taking the derivative of (5.50) we get:

$$\dot{x}(t) = -\frac{1}{h} \sum_{i=0}^{k} i q_i \left(\frac{t_{n+1} - t}{h} \right)^{i-1} , \tag{5.51}$$

which gives, evaluated at $t = t_{n+1}$,

$$\dot{x}_{n+1} = q_0(0) + q_1(-\frac{1}{h}) + q_2(0) + \ldots + q_k(0). \tag{5.52}$$

On the other hand, replacing the definition (5.50) in the backward differentiation formula (5.43) we have:

$$\dot{x}_{n+1} = -\frac{\alpha_0}{h} q_0 - \frac{\alpha_1}{h} \left[q_0 + q_1 \left(\frac{t_{n+1} - t_n}{h} \right) + \ldots + q_k \left(\frac{t_{n+1} - t_n}{h} \right)^k \right] - \ldots$$

$$-\frac{\alpha_k}{h} \left[q_0 + q_1 \left(\frac{t_{n+1} - t_{n+1-k}}{h} \right) + \ldots + q_k \left(\frac{t_{n+1} - t_{n+1-k}}{h} \right)^k \right] . \tag{5.53}$$

Grouping the q terms in 5.53:

$$\dot{x}_{n+1} = q_0 \left[-\frac{\alpha_0}{h} - \frac{\alpha_1}{h} - \ldots - \frac{\alpha_k}{h} \right] +$$

$$q_1\left[-\frac{\alpha_1}{h}\left(\frac{t_{n+1}-t_n}{h}\right)-\ldots-\frac{\alpha_k}{h}\left(\frac{t_{n+1}-t_{n+1-k}}{h}\right)\right]+\ldots$$

$$+q_k\left[-\frac{\alpha_1}{h}\left(\frac{t_{n+1}-t_n}{h}^k\right)-\ldots-\frac{\alpha_k}{h}\left(\frac{t_{n+1}-t_{n+1-k}}{h}^k\right)\right]. \qquad (5.54)$$

Equating the q_i coefficients in (5.52) and (5.54), a linear system can be obtained:

$$\begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & \left(\frac{t_{n+1}-t_n}{h}\right) & \left(\frac{t_{n+1}-t_{n-1}}{h}\right) & \ldots & \left(\frac{t_{n+1}-t_{n+1-k}}{h}\right) \\ 0 & \left(\frac{t_{n+1}-t_n}{h}\right)^2 & \left(\frac{t_{n+1}-t_{n-1}}{h}\right)^2 & \ldots & \left(\frac{t_{n+1}-t_{n+1-k}}{h}\right)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \left(\frac{t_{n+1}-t_n}{h}\right)^k & \left(\frac{t_{n+1}-t_{n-1}}{h}\right)^k & \ldots & \left(\frac{t_{n+1}-t_{n+1-k}}{h}\right)^k \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \qquad (5.55)$$

The α set so calculated gives the *exact* value $\dot{x}_{n+1}$ whenever $x(t)$ is any polynomial of degree $0, 1, \ldots, k$. If a constant step size h is used, (5.55) may be written in the simpler form

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 2 & 3 & \ldots & k \\ 0 & 1 & 4 & 9 & \ldots & k^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2^k & 3^k & \ldots & k^k \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \qquad (5.56)$$

5.6.2 Determination of the Jacobian

In the particular case of equation (5.1), the BDF algorithm leads to

$$\left(-\frac{\alpha_0}{h}\mathbf{C}_{n+1}+\mathbf{K}_{n+1}\right)\mathbf{x}_{n+1}+\left[-\frac{1}{h}\mathbf{C}_{n+1}\sum_{i=1}^{k}\alpha_i\mathbf{x}_{n+1-i}+\mathbf{f}_{n+1}\right]=\mathbf{0}. \qquad (5.57)$$

Equation (5.57) is nonlinear algebraic, and can be solved by a Newton iteration, in which case the Jacobian matrix must be calculated. Two cases are possible and we consider them next.

Derivatives of matrix–vector products are available

If the derivative of the matrix–vector products in (5.57) is available, we can write

$$\mathbf{J}_{cx} = \frac{\partial}{\partial \mathbf{x}} \mathbf{C}(\mathbf{x})\mathbf{x}, \tag{5.58}$$

$$\mathbf{J}_{kx} = \frac{\partial}{\partial \mathbf{x}} \mathbf{K}(\mathbf{x})\mathbf{x} \tag{5.59}$$

and the total Jacobian for the nonlinear equation (5.57) is

$$\mathbf{J} = -\frac{\alpha_0}{h}\mathbf{J}_{cx} + \mathbf{J}_{kx}. \tag{5.60}$$

If the damping matrix $\mathbf{C}$ is constant, then this latter expression becomes

$$\mathbf{J} = -\frac{\alpha_0}{h}\mathbf{C} + \mathbf{J}_{kx}. \tag{5.61}$$

Derivatives of matrices are available

If only the matrix derivatives are available, then we write

$$\mathbf{J}_c = \frac{\partial}{\partial \mathbf{x}} \mathbf{C}(\mathbf{x}), \tag{5.62}$$

$$\mathbf{J}_k = \frac{\partial}{\partial \mathbf{x}} \mathbf{K}(\mathbf{x}). \tag{5.63}$$

We begin by noting that the structure of (5.57) is the familiar stiffness matrix-load vector system of a static nonlinear problem and can be written equivalently as

$$\mathbf{F}(\mathbf{a}) = \mathbf{K}'\mathbf{a} + \mathbf{f}' = \mathbf{0}, \tag{5.64}$$

where $\mathbf{a} = \mathbf{x}_{n+1}$. The element in row i and column j of the Jacobian is given by

$$J_{ij} = \frac{\partial F_i}{\partial a_j} = \frac{\partial}{\partial a_j} \left[\mathbf{K}_i' \mathbf{a} + f_i'\right]$$

$$= K_{ij}' + \frac{\partial \mathbf{K}_i'}{\partial a_j}\mathbf{a} + \frac{\partial f_i'}{\partial a_j}. \tag{5.65}$$

Applying this result to (5.57) we arrive at

$$J_{ij} = \left(-\frac{\alpha_0}{h} C_{ij} + K_{ij}\right) + \left(-\frac{\alpha_0}{h}\frac{\partial \mathbf{C}_i}{\partial a_j} + \frac{\partial \mathbf{K}_i}{\partial a_j}\right)\mathbf{a} + \left(-\frac{1}{h}\sum_{l=1}^{k} \alpha_l x_{n+1-l}\right)\frac{\partial C_{ij}}{\partial a_j} + \frac{\partial f_i}{a_j}. \tag{5.66}$$

An important special case arises when the $\mathbf{C}$ matrix is constant, in which case the Jacobian (5.66) simplifies to

$$J_{ij} = \left(-\frac{\alpha_0}{h} C_{ij} + K_{ij}\right) + \frac{\partial \mathbf{K}_i}{\partial a_j}\mathbf{a} + \frac{\partial f_i}{a_j}. \tag{5.67}$$

5.6.3 Estimation of the next value

The nonlinear iteration requires a starting point $\mathbf{x}_{n+1}^{(0)}$ from which estimate a new value $\mathbf{x}_{n+1}^{(1)}$ and so on; this initial value may well be just the last calculated value $\mathbf{x}_n$.

However, it is known that convergence is more rapidly achieved if the initial guess is close to the true value. A polynomial approximation similar to the one used in the preceding section may be employed to provide an initial guess that is theoretically close to the solution. The idea is to fit a polynomial of order k over the $k+1$ points $t_{n-k}, t_{n-k+1}, \ldots, t_n$, that is, a polynomial of the same order to the one used for approximating the derivative, but displaced backwards in time. Evaluating it in $t = t_{n+1}$, we obtain an estimate of the true value $x(t_{n+1})$ since both polynomials share k points. Starting again from (5.50), we see that

$$x_{n+1} = q_0(1) + q_1(0) + q_2(0) + \ldots + q_k(0). \tag{5.68}$$

Now, as in the case of eq.(5.43), we define

$$x^P_{n+1} = \sum_{i=1}^{k+1} \gamma_i x_{n+1-i}, \tag{5.69}$$

where the superindex P indicates that this expression is to be used to predict the value $x(t_{n+1})$. Substituting (5.50) into (5.69) we get

$$x^P_{n+1} = \gamma_1 \left[q_0 + q_1 \left(\frac{t_{n+1} - t_n}{h} \right) + \ldots + q_k \left(\frac{t_{n+1} - t_n}{h} \right)^k \right]$$

$$+ \ldots + \gamma_{k+1} \left[q_0 + q_1 \left(\frac{t_{n+1} - t_{n-k}}{h} \right) + \ldots + q_k \left(\frac{t_{n+1} - t_{n-k}}{h} \right)^k \right]. \tag{5.70}$$

We then equate the q_i coefficients in (5.68) and (5.70) to obtain the linear system

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \left(\frac{t_{n+1}-t_n}{h}\right) & \left(\frac{t_{n+1}-t_{n-1}}{h}\right) & \cdots & \left(\frac{t_{n+1}-t_{n-k}}{h}\right) \\ \left(\frac{t_{n+1}-t_n}{h}\right)^2 & \left(\frac{t_{n+1}-t_{n-1}}{h}\right)^2 & \cdots & \left(\frac{t_{n+1}-t_{n-k}}{h}\right)^2 \\ \vdots & \vdots & \ddots & \vdots \\ \left(\frac{t_{n+1}-t_n}{h}\right)^k & \left(\frac{t_{n+1}-t_{n-1}}{h}\right)^k & \cdots & \left(\frac{t_{n+1}-t_{n-k}}{h}\right)^k \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (5.71)$$

If a constant step size h is used, (5.71) may be written in the simpler form

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & k+1 \\ 1 & 4 & 9 & \cdots & (k+1)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^k & 3^k & \cdots & (k+1)^k \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (5.72)$$

5.6.4 Implementation

The time stepping algorithm using the backward differentiation formula was implemented. The time discretization is made at the element level so that for each element, starting from (5.14) we end up with an expression like (5.57). This is a nonlinear algebraic equation which is itself transformed to a Newton equation. Thus, two transformations are performed and the result is a global linear equation; the whole process is repeated until convergence is achieved. The advantage of performing the transformations at the element level is that the assembler and linear equation solver do not change for linear, nonlinear, static and transient problems.

The boundary conditions can be handled in two ways: known values of a degree of freedom can be used to eliminate the corresponding equation, or the equation written and the known value incorporated by algebraic manipulation. In the first

case, the known values are passed to the right hand side of the final equation by the assembly subroutine; the BDF and Newton transformations are thus transparent and have no effect as long as the known values are concerned. In the second case, the algebraic manipulation must be made at the global equation level, so that the only requirement is that the transformations are made before incorporating the boundary conditions. This is guaranteed if the transformations are made at the element level.

As a test case, we use the inductor shown in Fig.5.12, which has a shell core made of nonconducting steel while the central leg was given a high conductivity. The coil is excited by an exponentially rising current and the magnetic vector potential is calculated. Fig.5.13 shows the results for two nodes placed in the central leg, one touching one of the coils and the other one half of the way towards the center; the results for the case where the conductivity is zero serve as a reference. In the case of nonzero conductivity, the diffusion effect is noticed, with the retarding effect being more marked for the node far from the coil.

The response at this latter node is studied using different time step sizes. Fig.5.14 shows the calculated response for time steps of 10 and 100ms. With the larger time step, the response is seen to contain a moderate error, but the economy of calculating less points may justify it. In fact, a still better alternative is to use higher order formulas with a large time step; the result of this approach is shown in Fig.5.15. In this particular case, the order is chosen to be the maximum allowable at each point; thus, the formula is first order for the first time step, second order for the second time step, etc., until a maximum order of four. As appreciated in the figure, the error is larger for the lower order formulas, but afterwards is very much reduced and as a result the calculations for the large time step case with higher order formulas are comparable to the results for a small time step.

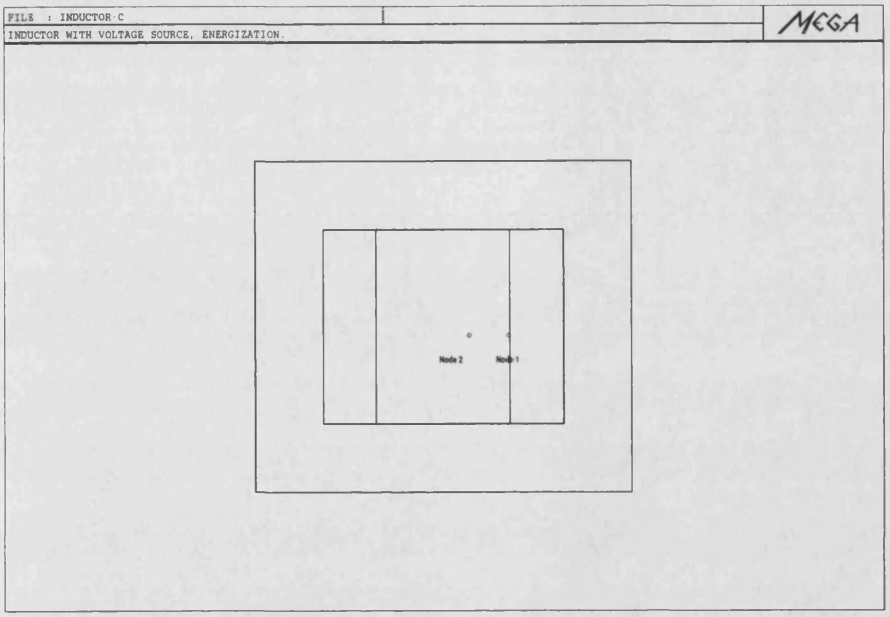


Figure 5.12: Inductor with conducting central leg and location of nodes.

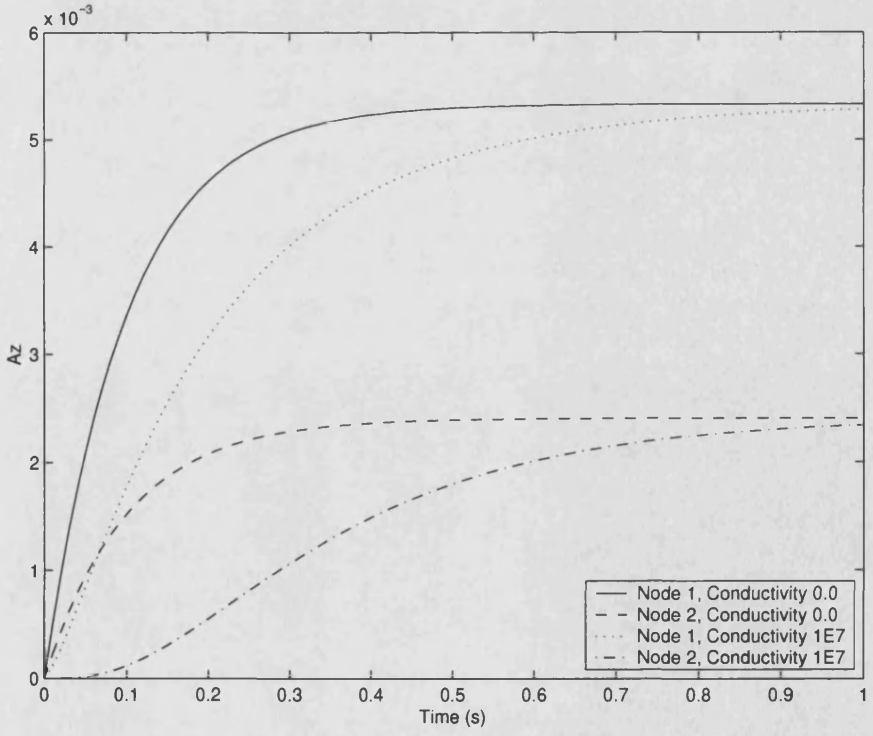


Figure 5.13: MVP for nodes 1 and 2, with and without conductivity. Time step is 10ms

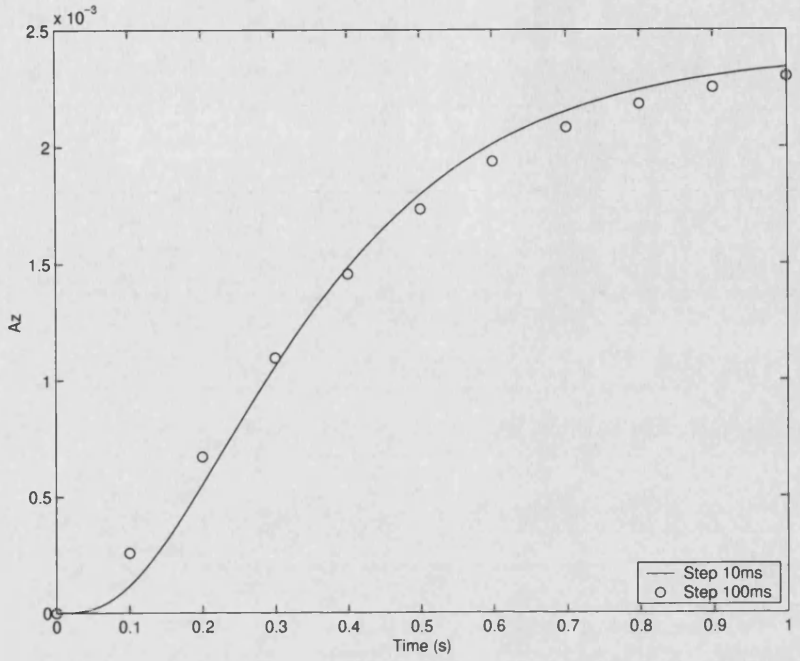


Figure 5.14: MVP at node 2 for different time step sizes.

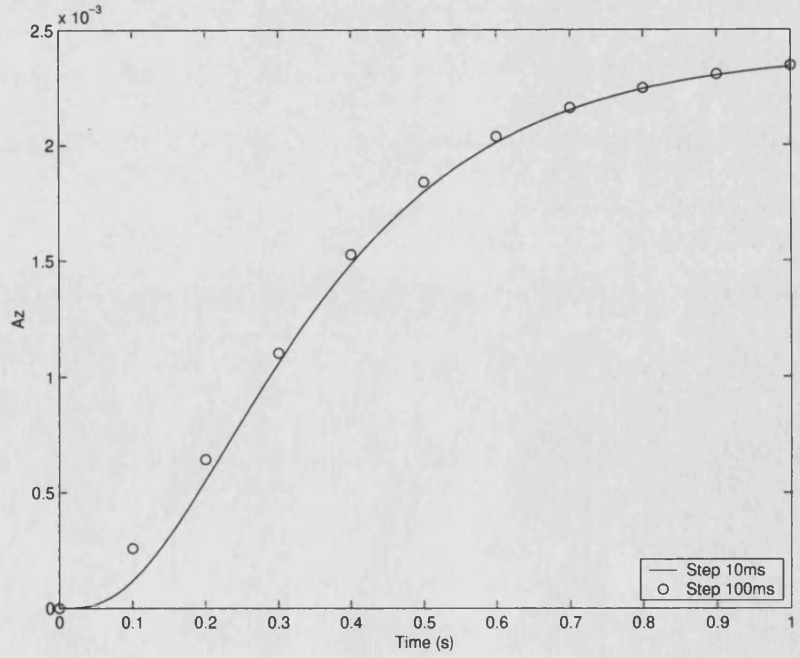


Figure 5.15: MVP at noded 2 for small first order and large higher order time steps.

5.6.5 Stiff problems

The results presented in the previous section use a constant time step; only the order is varied. A complete time step adaptation strategy has both varying order and varying time step size. We present a simple algorithm for that purpose. The equation for a van der Pol oscillator, a two-variable problem, is solved by a trusted implementation and used as a reference. The characteristics of the problem are studied first with constant time steps and then the step size varied.

The equation to solve is

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \mu(1 - x_i^2)x_2 - x_1. \tag{5.73}$$

Figs.(5.16) to (5.18) show x_1 for fixed step size $h = 0.5$, 0.1 and 0.05, with $\mathbf{x}_0 = [0, -2]$ using the first order formula, compared with the solution given by the variable-order, variable step-size code ODE15S [85]. The reference solution has 264 time steps. The BDF solution shown in Fig.(5.16) uses a time step size $h = 0.5$ (50 time steps) and has a large error both in phase and in magnitude. Reducing the time step to $h = 0.1$ (250 time steps) helps to alleviate the error in amplitude, at the cost of increasing the number of calculations, as seen in Fig.(5.17). Even Fig.(5.18) with $h = 0.05$ (500 time steps) shows an error in phase. The first order algorithm, which is non other than the backward Euler method, has the very desirable property of L-stability, as can be appreciated in Fig.(5.16), where even for a relatively large time step size, the solution remains bounded. This property has made this algorithm

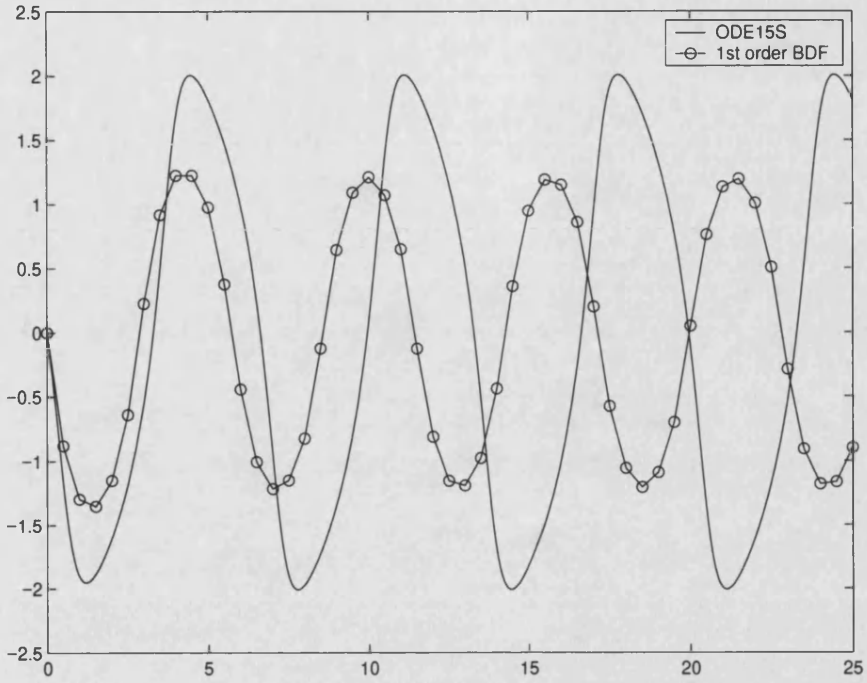


Figure 5.16: x_1 for $h = 0.5$, 50 steps.

a very popular one but, as we have seen, requires a relatively large number of time points to achieve a given accuracy.

Figs.(5.19) and (5.20) show the results for fixed step $h = 0.5$ and 0.1, for a second order BDF algorithm. The BDF solution shown in Fig.(5.19) uses a time step size $h = 0.5$ (50 time steps) and has an encouragingly small error in angle and magnitude (remember that the reference solution consists of 264 time steps). Reducing the time step to $h = 0.1$ (250 time steps) the BDF solution is virtually the same as the reference one.

In the case of a continuous problem discretized by the finite element method, where the solution for each time step requires considerable computing resources, a time stepping algorithm requiring the least number of points is very desirable. The BDF algorithm provides such alternative, in principle. It works in such a way that solves

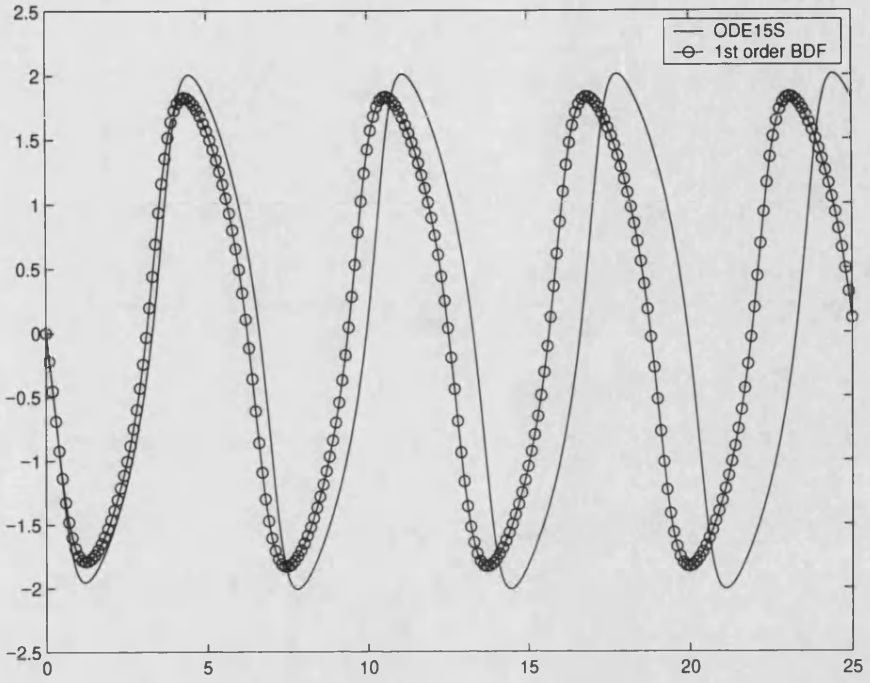


Figure 5.17: x_1 for $h = 0.1$, 250 steps.

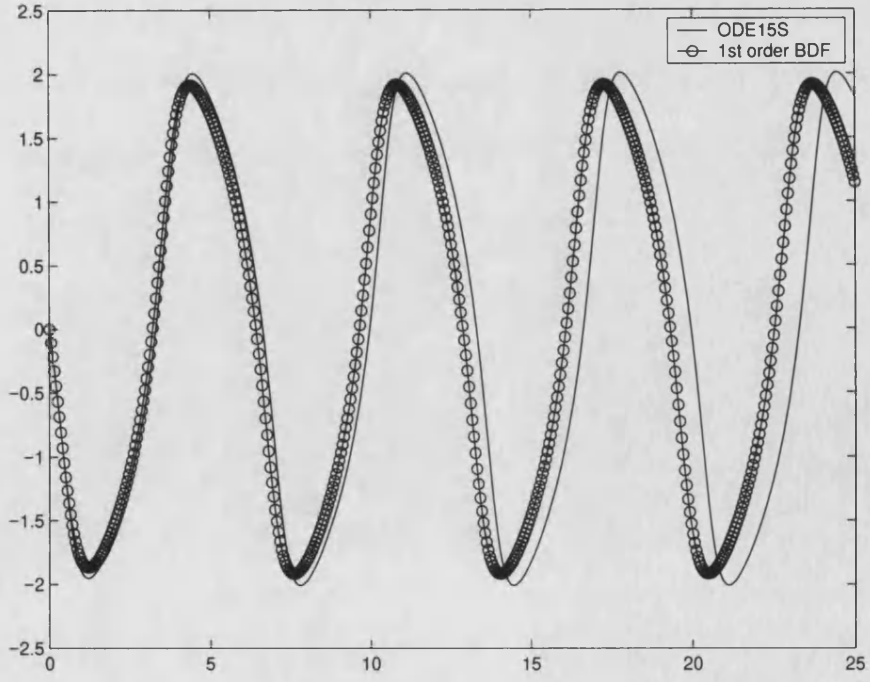


Figure 5.18: x_1 for $h = 0.05$, 500 steps.

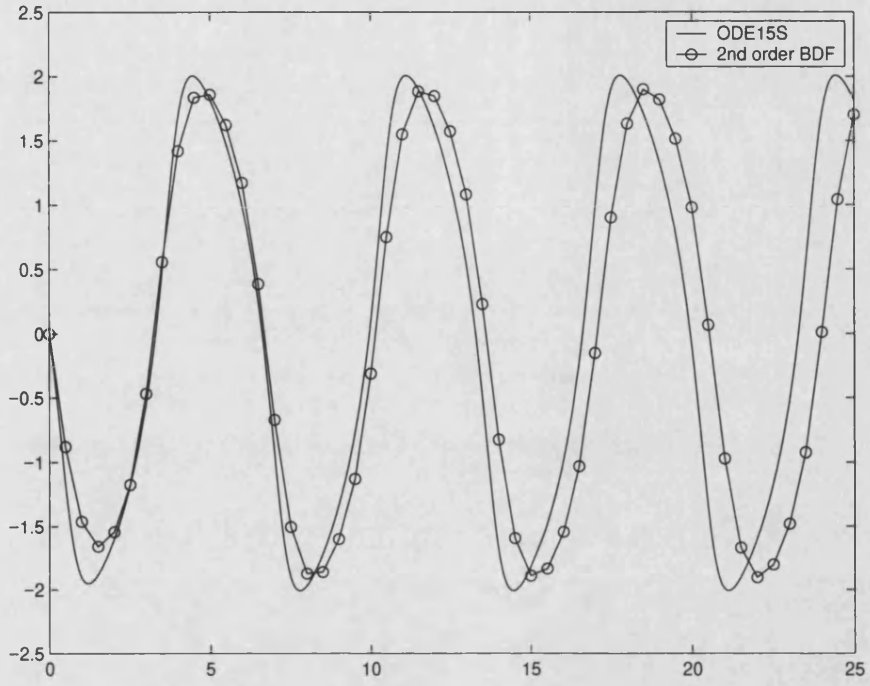


Figure 5.19: x_1 for $h = 0.5$, 50 steps.

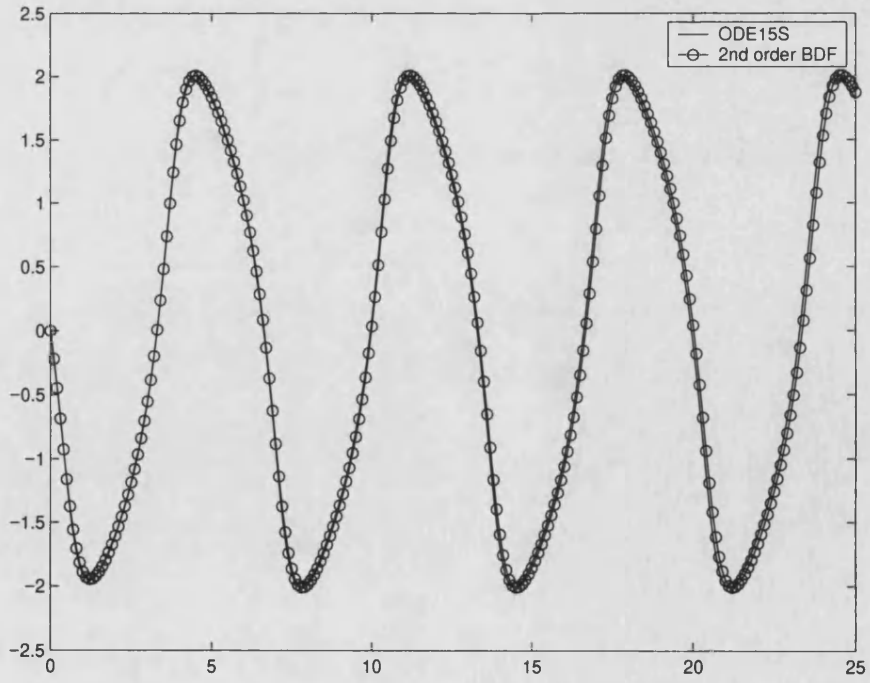


Figure 5.20: x_1 for $h = 0.1$, 250 steps.

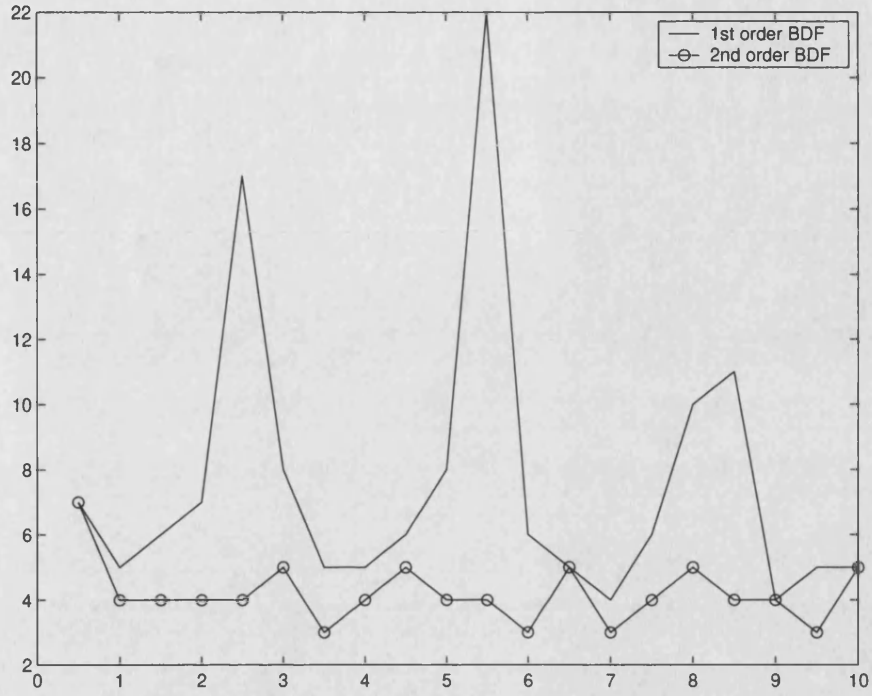


Figure 5.21: Number of Newton-Raphson iterations vs. Time (s), $h = 0.5$.

an implicit nonlinear equation at each time step, just the same as the backward-Euler method, but gives much more improved accuracy characteristics. It makes use of the solution at the last few time steps, but otherwise the process is not more expensive in computer time or memory than the first order methods. The key point is that, although the approximation is of higher-order, the solution process calculates one point at a time, thus making it equivalent to a first order method in this sense. Moreover, the BDF method can be easily augmented to give an estimation of the next state to be calculated, thus providing a convenient starting point for the Newton-Raphson iteration. This can be appreciated in Figs. 5.21 and 5.22, where the number of Newton-Raphson iterations for the first and second order BDF methods are presented for step sizes $h = 0.5$ and $h = 0.1$.

Fig. 5.23 compares the results for second and third order approximations, for the largest step size allowable while still keeping convergence. For a second order ap-

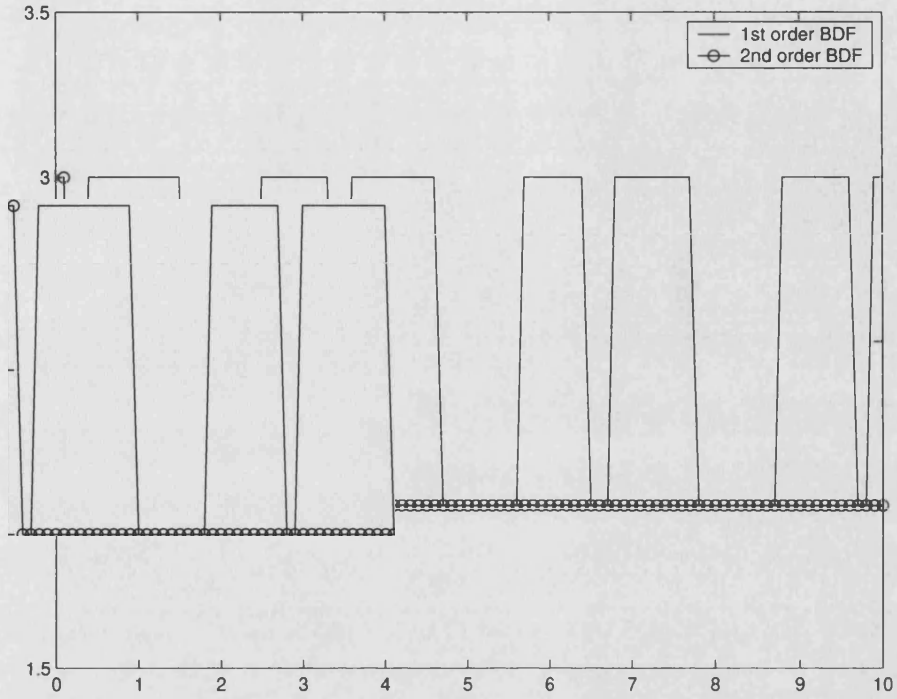


Figure 5.22: Number of Newton-Raphson iterations vs. Time (s), $h = 0.1$.

proximation, the maximum step size was $h = 0.7$; for an approximation of third order, $h = 0.85$. Although the error in phase is seen to be larger for the case of third order approximation, this is also seen to be due to the error introduced by the bad approximation near the points of rapid variation, where a smaller time step would have been be more appropriate.

These observations may be generalized, and thus it appears that the higher the order, the larger step sizes allowable, at least for regions of slow change. In practice, the order is limited to 5 or 4 because of stability reasons. The BDF formula of order 4 is stable in all cases, but the formula of order 5 is somewhat less stable and thus must be used carefully [85].

If the variables of the model have widely differing time constants, the problem is called "stiff". A problem of this kind is difficult to solve for it would require a very

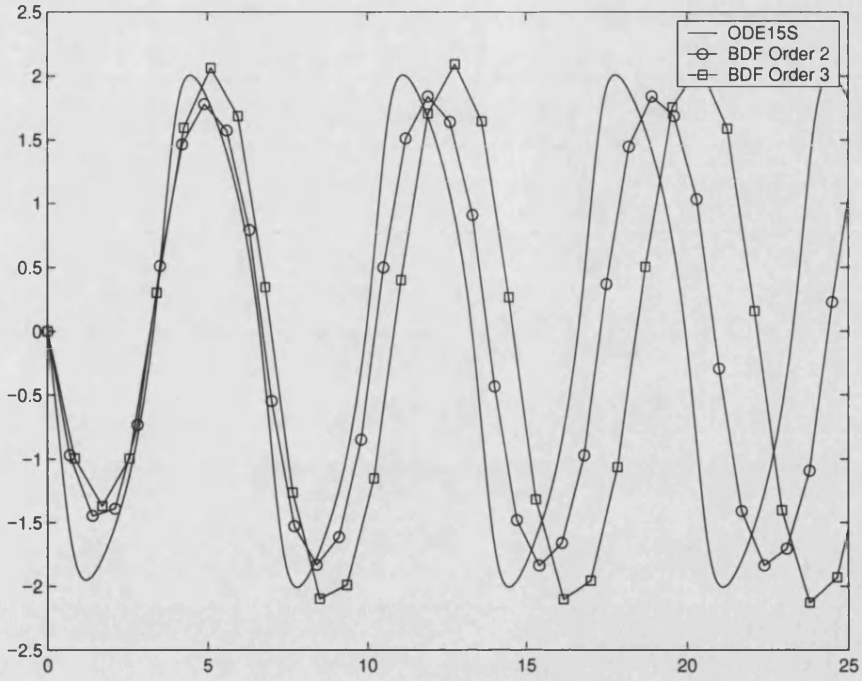


Figure 5.23: x_1 vs. time (s). Second and third order approximations.

small step size to handle the regions of rapid change, but allows a relatively big step size in some other regions. A constant step size is thus not convenient, as seen in Fig. 5.24 for the system of eq.(5.73), for $\mu = 100$ and $\mathbf{x}_0 = [2, 0.00667]$, where the BDF solution fails just before arriving to the point of abrupt change.

Another aspect to consider is the consistency of the initial conditions. An inconsistent state may be given at the start of the simulation, or the conditions of the problem change from one time step to another with the variables at the start of the Newton iteration taking the value of the previous converged step. The implicit time stepping algorithms have the property of "correcting" an inconsistent set of initial conditions. A fixed step algorithm will arrive to a consistent set of conditions in one time step, provided it converges; on the other hand, a variable step size algorithm will compensate for the inconsistency by requiring small step sizes. This can be appreciated by comparing Figs. 5.25 and 5.26; in the first case, the initial conditions

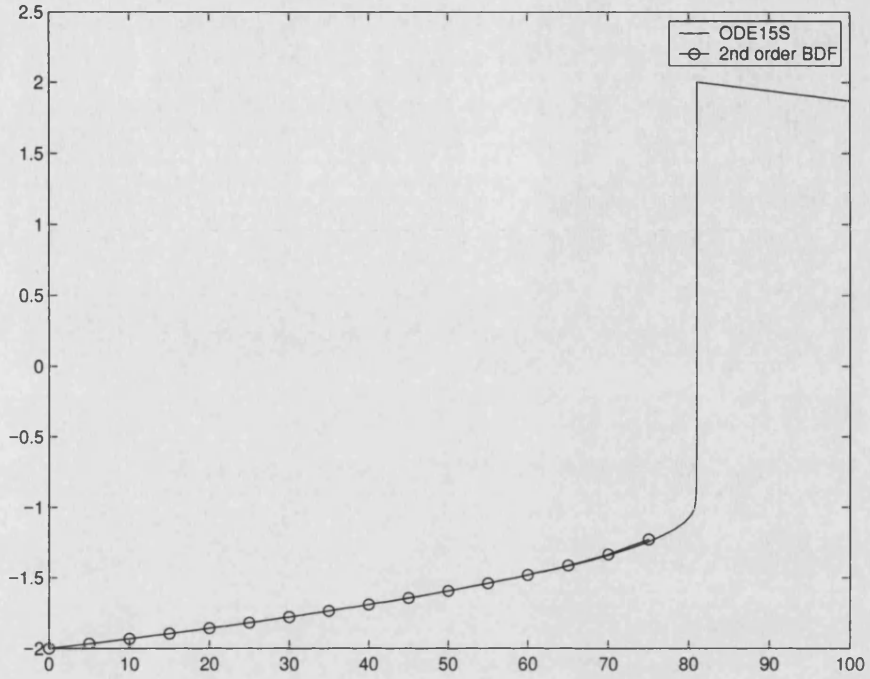


Figure 5.24: x_1 vs. time (s) for very stiff problem, μ =100. Large step size fails.

are consistent and a total of 22 steps are required, while the initial conditions for the second case are inconsistent, requiring a total of 82 steps.

Higher order methods are to be preferred over first order algorithms in order to allow bigger step sizes. However, in the case of stiff problems the step size must be adapted according to the changing conditions. This can be tricky, since the usual step adaptation algorithms tend to require large numbers of time points to ensure a given accuracy. Also, they generally require an absolute error to be specified for *each* variable.

5.6.6 A step adaptation algorithm

In the case of stiff problems, a fixed step algorithm will fail, either by being unable to find a convergent Newton-Raphson solution, or by giving wrong results. The

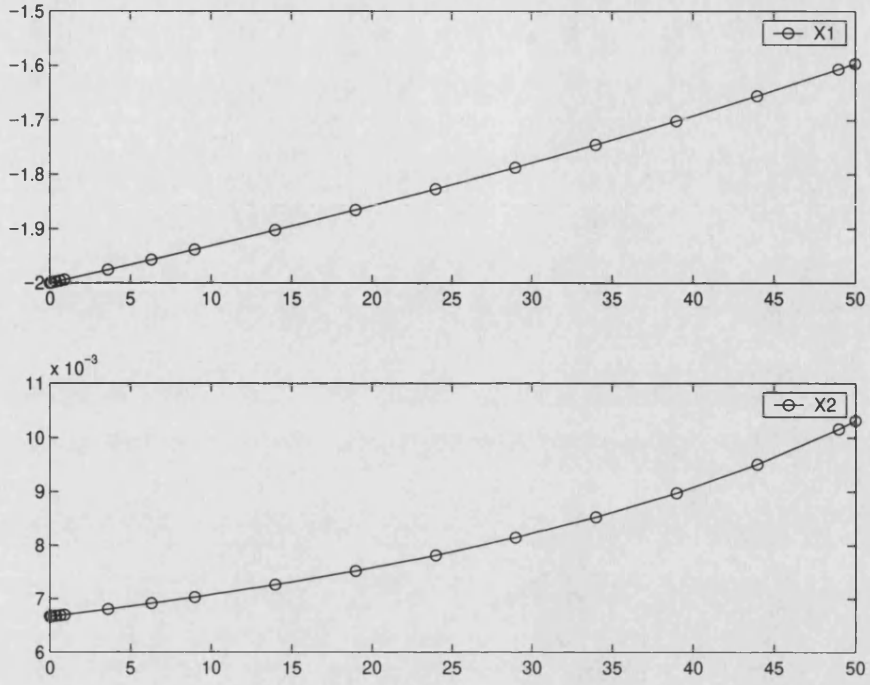


Figure 5.25: x_1 and x_2 vs time(s) with consistent initial conditions.

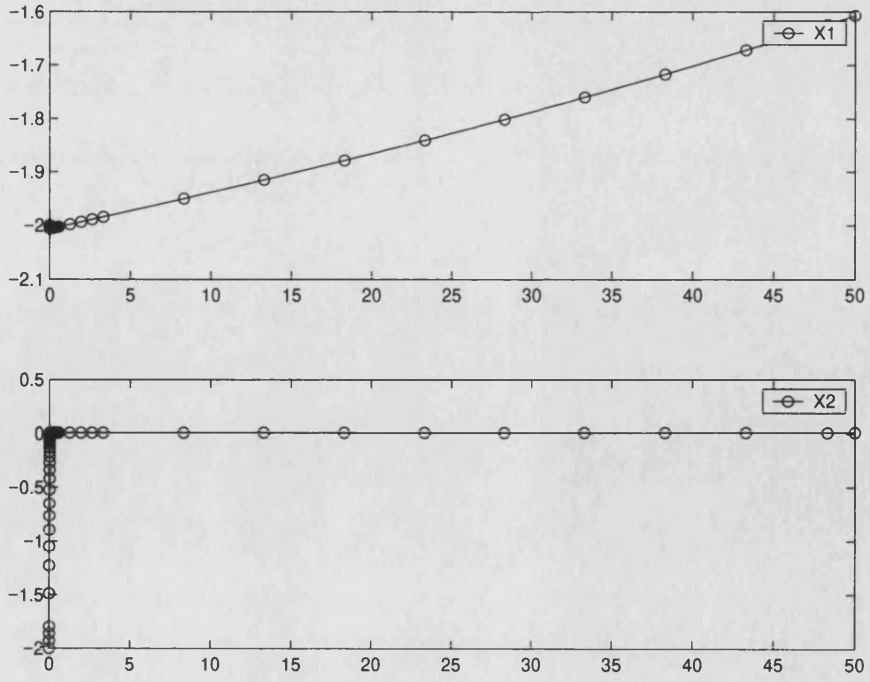


Figure 5.26: x_1 and x_2 vs time(s) with inconsistent initial conditions.

control of the error of the algorithm requires careful consideration. Here we present a particular implementation which gives acceptable results and requires only the specification of a global relative tolerance. The pseudo-code for the routine is:

```
0 - Set time span, initial step, reduction factor,
    increment factor,  maximum order, tolerance
while(  not finished )

  1 - Store state and advance time.
  2 - Predict next state using BDF beta formula.
  3 - Get BDF alpha coefficients at t(n+1).
  4 - Solve nonlinear equation.
    4.1 - Calculate error.
          Absolute error:
          e = h * ( x(n+1) - x_predicted ) / ( t(n+1) - t(n-k) )
          Relative error:
          e = log( abs( e ) ) - log( abs( x(n+1) ) )
    4.2 - Modify the step size if error is big.
      if( e > tolerance ) h = h_reduction * h
  5 - Set next step size.
    if( num_successes > k ) h = h_increment * h
  6 - Set order.
    k = min( [ num_points+1, kmax, num_successes+1] )
    if(rejected) k = 1

end
```

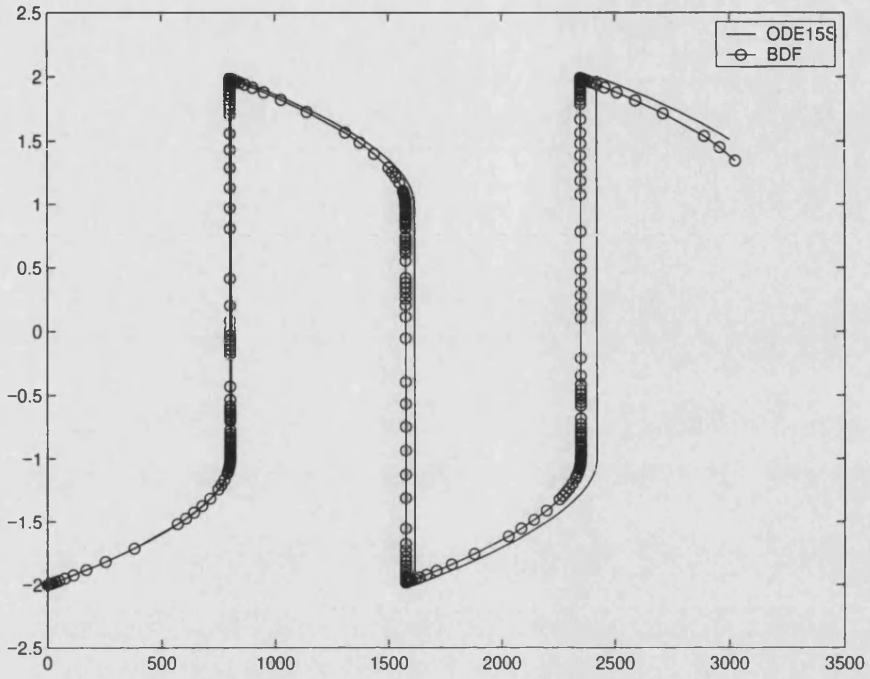


Figure 5.27: x_1 vs. time (s) for very stiff problem. 641 ODE15S steps, 668 BDF steps.

An error estimation formula is used to control the step size; if the relative error for a given time step is found to be larger than desired, the current calculation is rejected and the step size reduced. Similarly, if several contiguous time steps are successful, the step size is incremented. The reduction and increase in the step size are controlled by two constants, which were selected by experiment.

The time step is also reduced if the nonlinear solver fails to converge; however, since a prediction formula is used, this is a relatively rare condition. Fig. 5.27 shows the results for the system of eq.(5.73) with $\mu = 1000$ (a very stiff problem). In the case of the BDF algorithm, 210 steps were rejected (172 by error limit violation, and 38 by failed nonlinear convergence), and 668 accepted, for a total of 878 required to complete the simulation. The proportions are typical and in any simulation 25% of the calculated steps are expected to be rejected.

The order is incremented provided the sufficient number of past steps has been solved successfully, up to a maximum predefined order. If a given step fails to converge, the time step is reduced and the order reset to 1.

5.7 Conclusions

This chapter has been concerned with the treatment of the first order vector equation arising from finite element discretizations. Some frequently used time stepping algorithms have been considered; the classical Euler formulas were adapted to the vector case.

One central idea has been that the application of the time stepping formula at the element level offers several advantages not sufficiently emphasized in the literature. The first is that differential–algebraic systems of equations may be treated just as easily by conventional formulas; this was exemplified by the Adams–Moulton and Gear formulas of several orders adapted to our case. A second advantage is that the solution process is given a layered structure: the space discretization is made first, then the time discretization, and lastly the Newton transformation for the solution of the nonlinear algebraic equation. Since all this happens at the element level, the programming follows a logical sequence and the implementation is facilitated.

The frequently used θ-method was considered, and it was shown that some assumptions are implicit in the available expressions. The formula for constant matrices if used when such constraint is not met may lead to the algorithm losing stability. Expressions for the more general case of nonlinear equations rely on similar assumptions, in this case that the damping matrix is constant.

The very general method based on the backward differentiation formulas offers a convenient framework for the implementation of higher order formulas and step adaptation. The method has no restrictions about constant matrices; additionally, it is known to be specially well suited to handle stiff problems.

The various methods presented have a point of contact: the first order Adams–Moulton, Gear and BDF methods reduce to the familiar backward Euler formula. Moreover, the θ-method is very often used in practice with $\theta = 1$, which is again the backward Euler formula. The BDF approach provides an extension to higher orders and due to its flexibility it can be said that encompasses the other methods.

The BDF approach was applied to eddy current problems. It was found that a large time step can be used without loss of accuracy if higher order approximations are allowed. The scheme is completed by a time adaptation algorithm and we considered an implementation requiring only a global error tolerance to be specified.

Chapter 6

Conclusions

The work presented in this thesis has been concerned with the solution of coupled magnetic and mechanical systems. Techniques for the analysis of such systems have been reviewed and elaborated. One thread in the development has been the increased detail attainable from the simulation.

Lumped parameter models are the simplest and more efficient, and it has been shown how the amount of detail from a field solution can be effectively compressed into a model that reflects the complex interaction between nonlinear materials and inputs to the system. The use of reduced models results in significant savings in computer time; the simulations presented were typically two orders of magnitude faster than the corresponding field models.

Advancing one step in the level of detail, we considered electromechanical systems where the magnetic part was described in terms of fields. The advantage is that no approximations are made regarding losses in the device, namely eddy currents. The evolution of the system in most practical cases has two greatly differing time scales; the mechanical part is much slower in its response than the magnetic part. This has

the fortunate consequence that staggered schemes give adequate results in general, provided stable time-stepping schemes are used in both domains.

A third specialization of increased detail is the consideration of the distributed nature of the forces and displacements in the mechanical system. This requires the use of the equations of elasticity in addition to those of magnetism. The study of this kind of system was based on ferromagnetic plates, a subject which has received less attention than the deflection of conducting plates.

Two new schemes for model reduction based on tables were presented, one making use of the sum of the energy and coenergy of the device, and the other utilizing directly the magnetic vector potential to calculate the relevant energy-related quantities. This latter scheme can be extended to the case of devices with multiple coils.

The concept of interface variables was used for the solution of static problems with strong coupling. Also, the method of interlacing nested iterations was presented and applied.

The Maxwell stress tensor evaluated at the interface between a ferromagnetic material and the surrounding air was shown to provide a convenient way of calculating the surface force distribution. The concept was applied to the solution of plate problems.

Traditional finite difference time stepping methods were adapted to the first order system of equations arising from dynamical problems. The backward difference formula methodology was applied to eddy current problems and a simple scheme for step and order adaptation presented.

Bibliography

[1] H. W. Katz, editor. *Solid State Magnetic and Dielectric Devices.* John Wiley & Sons, London, 1959.

[2] G. Haertling. Ferroelectric ceramics: history and technology. *J.Am.Ceram.Soc.*, 82(4):797–818, 1999.

[3] P. Janker, M. Christmann, F. Hermle, T. Lorkowski, and S. Storm. Mechatronics using piezoelectric actuators. *Journal of the European Ceramic Society*, 19:1127–1131, 1999.

[4] M. Hoffman, T. Werle, R. Pfeiffer, and A. Binder. 2d and 3d numerical field computation of eddy-current brakes for traction. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 752–753, Sapporo, Japan, October 1999. Conference record.

[5] K. Ha, J. Hong, G. Kim, and D. Kang. The performance analysis of eddy current brake system. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 750–751, Sapporo, Japan, October 1999. Conference record.

[6] F. Moon. *Magneto-solid mechanics.* John Wiley & Sons, New York, 1984.

[7] S. Zhou. *Electrodynamic theory of superconductors.* Peter Peregrinus, London, 1991. IEE Electromagnetic Waves Series v.34.

[8] P. Rai-Choudhury, editor. *Handbook of Microlithography, Micromachining and Microfabrication.* SPIE Press monograph PM39 / IEE Material and Devices Series 12. SPIE Optical Engineering Press / The Institution of Electrical Engineers, Bellingham / London, 1997. Vol.1,2.

[9] D. A. Hall and C. E. Millar, editors. *Sensors and Actuators.* IOM Communications, London, 1999. Proceedings of a Conference held at the Manchester Business School, 15-16 July 1996.

[10] E. Quandt and A. Ludwig. Magneostrictive actuation in microsystems. *Sensors and Actuator A: Physical*, 81(1-3):275–280, April 2000.

[11] E. Kallenbach, H. Kube, V. Zoppig, K. Feindt, R. Hermann, and F. Beyer. New polarized electromagnetic actuators as integrated mechatronic components-design and application. *Mechatronics*, pages 769–784.

[12] H. Yajima, H. Wakiwaka, K. Minegishi, N. Fujiwara, and K. Takamura. Design of linear dc motor for high-speed positioning. *Sensors and Actuators A: Physical*, 81(1-3):281–284, April 2000.

[13] D. Howe. Magnetic actuators. *Sensors and Actuators A: Physical*, 81(1-3):268–274, April 2000.

[14] Y. Kawase, H. Kikuchi, and S. Ito. 3d nonlinear transient analysis of dynamic behavior of the clapper type dc electromagnet. *IEEE Transactions on Magnetics*, 27(5):4238–4241, September 1991.

[15] G. Bedrosian. A new method for coupling finite element field solutions with external circuits and kinematics. *IEEE Transactions on Magnetics*, 29(2):1664–1668, March 1999.

[16] P. Eustache, G. Meunier, and J. L. Coulomb. Finite element toolbox for generic coupling (magnetic, thermal,etc.). *IEEE Transactions on Magnetics*, 32(3):1461–1464, May 1996.

[17] C. Bailey, G. A. Taylor, M. Cross, and P. Chow. Discretisation procedures for multi-physics phenomena. *Journal of Computational and Applied Mathematics*, 103(1):3–17, 1999.

[18] M. G. Pantelyat. Magneto-thermo-mechanical behavior of ferromagnetic materials in impulse electromagnetic fields. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 580–581, Sapporo, Japan, October 1999. Conference record.

[19] K. Tani, T. Yamada, and Y. Kawase. Dynamic analysis of linear oscillatory actuator driven by voltage source using FEM with edge elements and 3d mesh coupling method. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 418–419, Sapporo, Japan, October 1999. Conference record.

[20] T. Hirasawa, H. Kotera, Y. Sakamoto, and S. Shima. A study of dynamic response of micro-actuator. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 420–421, Sapporo, Japan, October 1999. Conference record.

[21] P. Kuo-Peng, N. Sadowski, J. P. A. Bastos, and N. J. Batistela. Coupled field and circuit analysis considering the electromagnetic device motion. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 484–485, Sapporo, Japan, October 1999. Conference record.

[22] Z. Ren and A. Razek. A strong coupled model for analysing the dynamic behaviour of non-linear electromechanical systems. *IEEE Transactions on Magnetics*, 30(5):3252–3255, September 1994.

[23] T. Takagi and J. Tani. Dynamic behavior analysis of a plate in magnetic field by full coupling and mmd methods. *IEEE Transactions on Magnetics*, 30(5):3296–3299, September 1994.

[24] S. Niikura and A. Kameari. Analysis for coupled problems between eddy currents and dynamic deflections of a thin shell structure. *IEEE Transactions on Magnetics*, 30(5):3284–3287, September 1994.

[25] Y. Crutzen, S. Papadopoulos, and F. Van Paemel. Modelling, analysis and validation of electromagnetic-mechanical coupling on clean test pieces. *IEEE Transactions on Magnetics*, 30(5):3276–3279, September 1994.

[26] Z. Ren, B. Ionescu, M. Besbes, and A. Razek. Calculation of mechanical deformation of magnetic materials in electromagnetic devices. *IEEE Transactions on Magnetics*, 31(3):1873–1876, May 1995.

[27] M. Besbes, Z. Ren, and A. Razek. Finite element analysis of magneto-mechanical coupled phenomena in magnetostrictive materials. *IEEE Transactions on Magnetics*, 32(3):1058–1061, May 1996.

[28] J. R. Brauer, J. J. Ruehl, M. A. Juds, M. J. VanderHeiden, and A. A. Arkadan. Dynamic stress in magnetic actuator computed by coupled structural and electromagnetic finite elements. *IEEE Transactions on Magnetics*, 32(3):1046–1049, May 1996.

[29] A. A. Arkadan, M. J. Vander Heiden, and J. R. Brauer. The coupled problem in electromagnetic ac contactors. *IEEE Transactions on Magnetics*, 33:1630–1633, March 1997.

[30] H. H. Woodson and J. R. Melcher. *Electromechanical Dynamics Vol.I: Discrete systems.* John Wiley & Sons, New York, 1968.

[31] P. I. Koltermann, J. P. A. Bastos, and S. R. Arruda. A model for dynamic analysis of AC contactor. *IEEE Transactions on Magnetics*, 28(2):1348–1350, March 1992.

[32] H. Martin-Gutierrez and P. I. Ro. Parametric modeling and control of a long-range actuator using magnetic servo levitation. *IEEE Transactions on Magnetics*, 34(5):3689–3695, September 1998.

[33] T. Soderstrom and P. Stoica. *System Identification.* Systems and Control Engineering. Prentice Hall, New York, 1989.

[34] A. H. Al-Khoury and J. K. Sykulski. Field modelling and characteristics simulation in electromechanical transducers. *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 17(1/2/3):146–150, 1998.

[35] H. M. Park and D. H. Cho. The use of the Karhunen-Loeve decomposition for the modeling of distributed parameter systems. *Chemical Engineering Science*, 51(1):81–98, 1996.

[36] I. Muntenau, T. Wittig, T. Weiland, and D. Ioan. Circuit parameter extraction for general electromagnetic devices. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 492–493, Sapporo, Japan, October 1999. Conference record.

[37] E. S. Hung and S. D. Senturia. Generating efficient dynamical models for micro-electromechanical systems from a few finite-element simulation runs. *Journal of Microelectromechanical Systems*, 8(3):280–289, September 1999.

[38] L. Bompa, P. Schueller, and J. C. Sabonnadiere. Analysis and synthesis of an electromagnet used for circuit breaker operation. *IEEE Transactions on Magnetics*, 21(6):2464–2467, November 1985.

[39] P. S. Sangha. *Transient Solution of Switched Devices Using Finite Elements.* PhD thesis, University Of Bath, 1991.

[40] C. Cereda, C. Gemme, and C. Reuber. Synchronous MV circuit-breaker with magnetic drive and electronic control. *ABB Review*, (6):13–21, 1999.

[41] H. C. Roters. *Electromagnetic Devices.* John Wiley & Sons, London, 1941.

[42] B. Aldefeld. A numerical solution of transient nonlinear eddy-current problems including moving iron parts. *IEEE Transactions on Magnetics*, 14(5):371–373, September 1978.

[43] F. Henrotte, A. Nicolet, H. Hedia, A. Genon, and W. Legros. Modelling of electromechanical relays taking into account movement and electric circuits. *IEEE Transactions on Magnetics*, 30(5):3236–3239, September 1994.

[44] H. C. Lai, D. Rodger, and P. C. Coles. A finite element scheme for colliding meshes. *IEEE Transactions on Magnetics*, 35(3):1362–1364, May 1999.

[45] H. C. Lai, P. C. Coles, D. Rodger, and P. J. Leonard. Transient analysis of an electromagnetic actuator using an overlapping finite element scheme. *IEEE Transactions on Magnetics*, 36(4):1462–1467, July 2000.

[46] D. C. White and H. H. Woodson. *Electromechanical Energy Conversion.* John Wiley & Sons, New York, 1959.

[47] J. Meisel. *Principles of Electromechanical Energy Conversion.* Electrical and Electronic Engineering Series. McGraw-Hill, New York, 1966.

[48] Z. Ren and A. Razek. Modelling of dynamical behaviour of electro-magneto-mechanical coupled systems. In *Proceedings of the IEE Second International Conference on Computation in Electromagnetics*, April 1994. Conference Publication Number: 384.

[49] K. Srairi, M. Feliachi, and Z. Ren. Electromagnetic actuator behaviour analysis using finite element and parametrisation methods. *IEEE Transactions on Magnetics*, 31(6):3497–3499, November 1995.

[50] T. E. McDermott, P. Zhou, J. Gilmore, and Z. Cendes. Electromechanical system simulation with models generated from finite element solutions. *IEEE Transactions on Magnetics*, 33:1682–1685, March 1997.

[51] A. Canova, M. Otella, and D. Rodger. A coupled field-circuit approach to 3D FEM analysis of electromechanical devices. In *Ninth International Conference on Electrical Machines and Drives*, pages 71–75. IEE, 1999. Conference Publication No.468.

[52] R. Gollee, T. Roschke, and G. Gerlach. A finite element method-based dynamic analysis of a long-stroke linear actuator. *Journal of Magnetism and Magnetic Materials*, 197:943–945, 1999.

[53] R. Gollee and G. Gerlach. A FEM-based method for analysis of the dynamic behavior of ac contactors. In *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, pages 42–43, Sapporo, Japan, October 1999. Conference Record.

[54] Y. Weili, R. Liyun, Y. Qinxin, L. Yongchao, S. Xueqin, and W. Jingguo. Finite element dynamic analysis for the electromagnet of AC contactor. *IEEE Transactions on Magnetics*, 27:4133–4135, September 1991.

[55] S. Wang, T. Miyano, and M. Hubbard. Electromagnetic field analysis and dynamic simulation of a two-valve solenoid actuator. *IEEE Transactions on Magnetics*, 29(2):1741–1746, March 1993.

[56] N. Sadowski, R. Carlson, A. M. Beckert, and J. P. A. Bastos. Dynamic modeling of a newly designed linear actuator using 3D edge element analysis. *IEEE Transactions on Magnetics*, 32(3):1633–1636, May 1996.

[57] N. A. Demerdash and T. W. Nehl. Electric machinery parameters and torques by current and energy perturbations from field computations - Part I: Theory and formulation. *IEEE Transactions on Energy Conversion*, 14(4):1507–1513, December 1999.

[58] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method Vol.I: Basic Formulation and Linear Problems.* McGraw Hill, London, 1989. Fourth Edition.

[59] D. Rodger. Finite-element method for calculating power frequency 3-dimensional electromagnetic field distributions. *IEE Proceedings*, 130(5):233–238, July 1983.

[60] N. Allen. *Finite element modelling of eddy currents in non-linear and moving media.* PhD thesis, University of Bath, Bath, 1997.

[61] P. J. Leonard and D. Rodger. Some aspects of two and three dimensional transient eddy current modelling using finite elements and single step marching algorithms. *IEE Proceedings*, 135 Part A(3):159–166, March 1988.

[62] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method Vol.II: Solid and Fluid Mechanics, Dynamics and Non-linearity.* McGraw Hill, London, 1989. Fourth Edition.

[63] N. R. Aluru and J. White. A multilevel Newton method for mixed-energy domain simulation of MEMS. *Journal of Microelectromechanical Systems*, 8(3):299–307, September 1999.

[64] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19:577–593, 1965.

[65] Z. Johan, T. J. R. Hughes, and F. Shakib. A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids. *Computer Methods in Applied Mechanics and Engineering*, (87):281–304, 1991.

[66] S. Artlich and W. Mackens. Newton-coupling of fixed point iterations. In W. Hackbush and G. Wittum, editors, *Numerical treatment of coupled systems*, volume 51 of *Notes on Numerical Fluid Mechanics*, pages 1–10. Vieweg, Braunschweig, 1995. Proceedings of the 11th GAMM Seminar, Kiel, 1995.

[67] J. H. Prevost. Partitioned solution procedure for simultaneous integration of coupled-field problems. *Commun.Numer.Meth.Engng*, 13:239–247, 1997.

[68] M. Cervera, R. Codina, and M. Galindo. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engineering Computations*, 13(6):4–30, 1996.

[69] I. Doltsinis. Solution of coupled systems by distinct operators. *Engineering Computations*, 14(8):829–868, 1997.

[70] D. Rodger, N. Allen, H. C. Lai, and P. J. Leonard. Calculation of transient 3D eddy currents in nonlinear media - verification using a rotational test rig. *IEEE Transactions on Magnetics*, 30(5):2988–2991, September 1994.

[71] N. Allen and D. Rodger. Description of TEAM Workshop problem 24: Nonlinear time-transient rotational test rig. In *Proceedings of the TEAM Workshop in the Sixth Round*, pages 57–60, Okayama, Japan, March 1996.

[72] A. Kovetz. *Electromagnetic Theory.* Oxford University Press, Oxford, 2000.

[73] H. H. Woodson and J. R. Melcher. *Electromechanical Dynamics, Vol.II: Fields, forces and motion.* John Wiley & Sons, New York, 1968.

[74] J. M. Gere and S. P. Timoshenko. *Mechanics of Materials.* PWS, Boston, 1997. Fourth Edition.

[75] I. M. Smith and D. V. Griffiths. *Programming the Finite Element Method.* John Wiley & Sons, Chichester, second edition edition, 1988.

[76] J. E. Akin. *Finite Elements for Analysis and Design.* Computational Mathematics and Applications. Academic Press, London, 1994.

[77] D. Hitchings. *Linear Statics Benchmarks.* NAFEMS, Glasgow, November 1987. Document LSB1, LSB2.

[78] D. Rodger, P. J. Leonard, and H. C. Lai. Surface elements for modelling 3D fields around thin iron sheets. *IEEE Transactions on Magnetics*, 29(2):1483–1486, March 1993.

[79] T. S. Parker and L. O. Chua. *Practical Numerical Algorithms for Chaotic Systems.* Springer-Verlag, New York, 1989.

[80] G. M. G. Hainsworth. *Design and analysis of pulsed induction accelerators.* PhD thesis, University of Bath, Bath, 1996.

[81] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall, Englewood Cliffs, 1971.

[82] L. O. Chua and P. Lin. *Computer-Aided Analysis of Electronic Circuits.* Prentice-Hall, Englewood Cliffs, 1975.

[83] A. Nicolet and F. Delince. Implicit Runge–Kutta methods for transient magnetic field computation. *IEEE Transactions on Magnetics*, 32(3):1405–1408, May 1996.

[84] I. A. Tsukerman, A. Konrad, G. Bedrosian, and M. V. K. Chari. A survey of numerical methods for transient eddy current problems. *IEEE Transactions on Magnetics*, 29(2):1711–1716, March 1993.

[85] L. F. Shampine and M. W. Reichelt. The Matlab ODE suite. *SIAM J.Sci.Comput.*, 18(1):1–22, January 1997.

[86] O. C. Zienkiewicz and W. L. Wood. Transient response analysis. In H. Kardestuncer, D. H. Douglas, and F. Brezzi, editors, *Finite Element Handbook.* McGraw Hill, New York, 1987. Part 2, Chapter 8.

[87] K. E. Brenan and B. E. Engquist. Backward differentiation approximations of nonlinear differential/algebraic systems. *Mathematics of Computation*, 51(184):659–676, October 1988.

[88] C. W. Gear. Simultaneous numerical solution of differential-algebraic equations. *IEEE Transactions on Circuit Theory*, 18(1):89–95, January 1971.

[89] R. K. Brayton, F. G. Gustavson, and G. Hachtel. A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas. *Proceedings of the IEEE*, 60(1):98–108, January 1972.

[90] O. C. Zienkiewicz, J. P. Gago, and D. W. Kelly. The hierarchical concept in finite element analysis. *Computers and Structures*, 16(1-4):53–65, 1983.

[91] E. Melgoza, H. C. Lai, and D. Rodger. Interlaced nonlinear iteration for coupled problems. *IEEE Transactions on Magnetics*, September 2001.

[92] H. C. Lai, P. C. Coles, E. Melgoza, D. Rodger, P. J. Leonard, and R. J. Hill-Cottingham. Time transient analysis of a rotational test rig benchmark problem. In *Compumag*, Evian, July 2001.

[93] E. Melgoza and D. Rodger. Comparison of table models of electromagnetic actuators. In *Compumag*, Evian, July 2001.

[94] E. Melgoza. Coupling circuit simulators and field models of magnetic devices. In *Compumag*, Evian, July 2001.

Related publications

Interlaced Nonlinear Iteration for Coupled Problems [91].

Time Transient Dynamic Analysis of a Rotational Test Rig Benchmark Problem [92].

Comparison of Table Models of Electromagnetic Actuators [93].

Coupling Circuit Simulators and Field Models of Magnetic Devices [94].

Interlaced Nonlinear Iteration for Coupled Problems

E. Melgoza, *Member, IEEE*, H. C. Lai and D. Rodger

Abstract— **A fast–converging solution method with strong coupling, which does not require the calculation of a global Jacobian, is discussed. The method is applied to an equilibrium–type coupled electromechanical system and also to the more general problem of the dynamic evolution of the system.**

Keywords— **Actuators, Finite element methods, Modeling, Transient response**

I. Introduction

CONSIDER a system formed by an axisymmetric solenoid actuator, Fig. 1, whose moving member is restrained by a spring; the equilibrium position is to be determined. The electromagnetic and mechanical systems may be discretised separately using some suitable procedure, for instance the finite element method. The resulting equations have the form

$$\mathbf{K}_e\mathbf{x}_e + \mathbf{f}_e = \mathbf{0}, \tag{1}$$

$$\mathbf{K}_m\mathbf{x}_m + \mathbf{f}_m = \mathbf{0}, \tag{2}$$

where the subscripts e and m stand for 'electrical' and 'mechanical' subsystems, respectively; $\mathbf{K}$ is the system matrix, $\mathbf{f}$ is the source vector and $\mathbf{x}$ the unknown variable.

Several approaches have been proposed for the solution of coupled problems like the one specified by the system (1)-(2). The simplest of these consists in solving alternately the electromagnetic and mechanical equations, starting from a known initial state and stopping when a convergence criteria is met [1], [2], [3], [4]. If the equations are nonlinear, a nested nonlinear iteration is required in each domain, making the process expensive. We will refer to this approach as block Gauss–Seidel iteration. To alleviate its shortcomings, a Newton–Raphson iteration for the nonlinear system (1)-(2) has been used [5], [6]. The process deals with the nonlinearity and the coupling at the same time, achieving a fast convergence. However, some drawbacks exist here too, among them the need to calculate the Jacobian matrix, the increased complexity of the implementation and the augmented size of the problem. Further variations of the Newton–Raphson iteration have been proposed addressing those concerns, for instance the use of matrix–free techniques [7], [8], [9]. Because the coupled–system equations are satisfied to a specified tolerance in both the block Gauss–Seidel and Newton–Raphson processes, they are said to provide a 'strong coupling' between physical domains.

Manuscript received June 5, 2000. This work was supported in part by CONACYT.

The authors are with the Applied Electromagnetics Research Centre, University of Bath. BATH, BA2 7AY, UK.

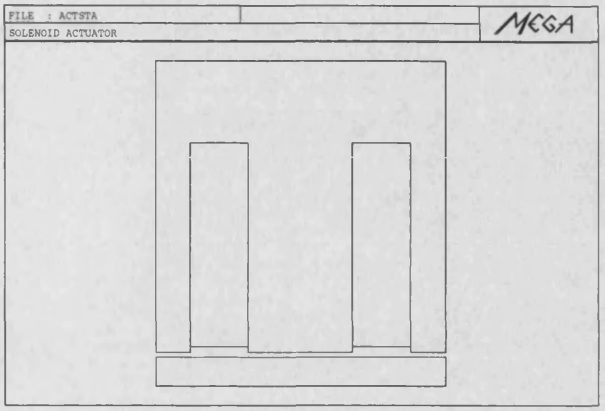


Fig. 1. Outline of the actuator.

II. Interlaced nonlinear iteration

A method for strong coupling combining the ease of implementation of the block iteration method with the fast convergence of the Newton–Raphson iteration is thus desirable. The convergence characteristics of the block iteration were investigated for our example problem. The electrical part was modelled by using finite elements and the mechanical part is modeled as a rigid body, thus giving only one equation. Table I shows the cumulative number of linear equation solutions for the electrical domain. However, consideration of the internal nonlinear equation solution reveals that improvements are possible. Fig. 2 shows the force for the internal Newton–Raphson process in the first block iteration. The force for the first and last iterations is almost the same, which suggests that full convergence is not required. We could perform *one* Newton–Raphson iteration of the electromagnetic part in the usual manner but, instead of waiting for the solution to converge for a given position, the mechanical equation is solved and the process repeated until the global variables converge. We shall refer to the outlined strategy as an interlaced iteration. This approach is well suited to the case where separate solvers exist for each domain, so that the implementation requires only a control program. In fact, access to the code is not required. Such an approach has been used by [2], [3].

Table II shows the convergence of the interlaced iteration for the test problem. Both approaches are compared in Fig. 3, which shows the position as a function of the number of linear equation solutions.

III. Dynamic problems

When the dynamic evolution of the system is of interest, the corresponding equations have the form

$$\mathbf{C}_e\dot{\mathbf{x}}_e + \mathbf{K}_e\mathbf{x}_e + \mathbf{f}_e = \mathbf{0}, \tag{3}$$

TABLE I

CONVERGENCE OF BLOCK ITERATION.

Iteration	Force(N)	Position(μm)	Lin.Eq.Solutions (cumulative)
1	13.1	503.6	4
2	95.5	526.5	10
3	112.1	531.1	14
4	115.9	532.2	16
5	116.8	532.4	17

TABLE II

CONVERGENCE OF INTERLACED ITERATION.

Iteration	Force(N)	Position(μm)	Lin.Eq.Solutions (cumulative)
1	12.3	503.4	1
2	96.5	526.8	2
3	112.5	531.2	3
4	116.0	532.2	4
5	116.8	532.4	5

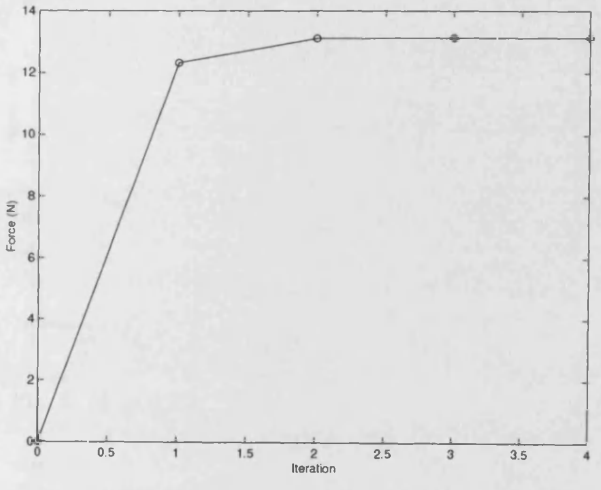


Fig. 2. Force for the first internal nonlinear solution.

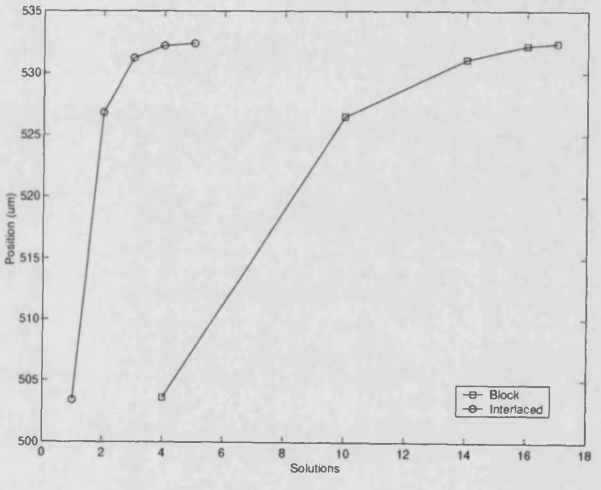


Fig. 3. Convergence of iteration schemes.

$$\mathbf{M}_m\ddot{\mathbf{x}}_m + \mathbf{C}_m\dot{\mathbf{x}}_m + \mathbf{K}_m\mathbf{x}_m + \mathbf{f}_m = \mathbf{0}. \tag{4}$$

After the time dimension is discretised, for example by using the backward Euler or the θ–method, the resulting equations are precisely in the form of (1)–(2), making the interlaced iteration process applicable in this case too. This contrasts with the frequently used approach of performing just one block Gauss–Seidel iteration, a method known as 'staggered', where convergence of the coupled–system equations is not pursued and the domains are said to be 'weakly coupled' [10]. Both approaches were used to simulate the closing of the armature of the actuator of Fig. 1, and compared against experimental data; the dimensions, material characteristics and experimental results are reported in [11].

The simulated transient consists in the closing of the armature loaded only by its own weight; the initial position is 0.0 mm and the final position is 0.73 mm. Fig. 4 compares the measured and calculated currents with a time step Δt of 10^{-2} s for both the staggered and the interlaced iteration; Fig. 5 shows the corresponding positions. The interval of armature movement is approximately 0.04 to 0.07 s; the electrical transient persists until about $t = 0.5$ s (not shown entirely). It can be appreciated that the interlaced process gives better results, as was to be expected since the coupling equation is solved consistently.

If a shorter time step is used, both the interlaced and the staggered solutions get closer to the measured values and very close to each other. For $\Delta t = 10^{-3}$ s, the currents are practically the same as seen in Fig. 6. A slight deviation in the position is observed in Fig. 7. The difference with respect to the measured values at the instant of closing is attributed to the elastic impact not being modeled.

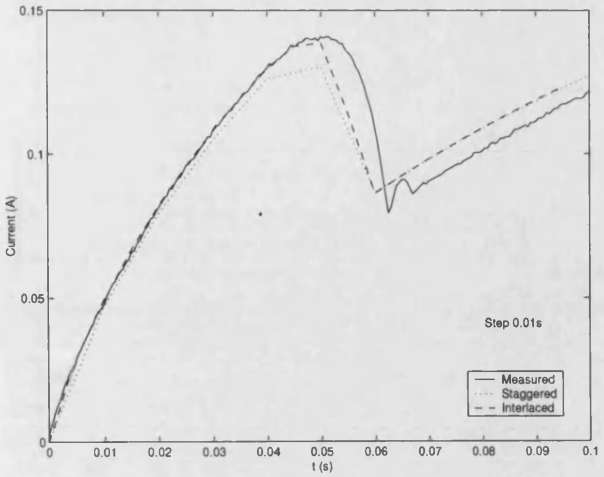


Fig. 4. Measured and simulated current with $\Delta t = 10^{-2}$ s.

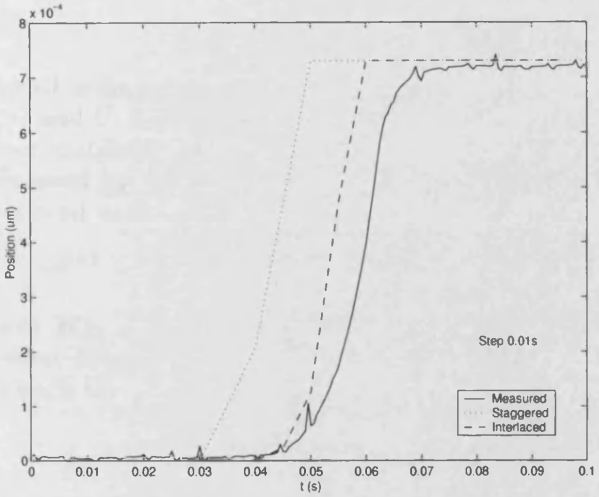


Fig. 5. Measured and simulated position with $\Delta t = 10^{-2}$ s.

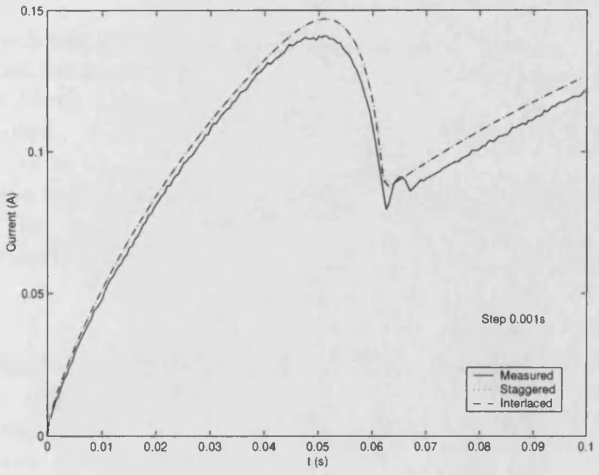


Fig. 6. Measured and simulated current with $\Delta t = 10^{-3}$ s.

IV. Computational cost

From the results discussed in the previous section it appears that the staggered solution, even if it does not rely on a consistent solution of the coupled equations, is equivalent to a strong coupling method, such as the interlaced iteration, when the time step is small enough. The reason is that for a small time step, the force is almost constant over a time step and along a small distance, so that it makes little difference if it is calculated at the beginning or at the end of the interval.

If the time stepping procedure uses a variable time step, then it is convenient to have strong coupling between domains in order to achieve better accuracy. If, on the other hand, a fixed, small time step is used by a constant–step algorithm, then it may not be worthwhile to provide a strong coupling. In order to assess the trade–offs in this respect, the computational cost of the strong coupling has to be determined.

In Fig. 8, the number of linear equation solutions (for the electrical part) is shown as a function of time, for a time step size of 10^{-3} s. The interlaced iteration requires more solutions of the linear equation; in fact, it requires approximately 20% more for our implementation, as seen in Table III. Thus, when a large and constant or a variable time step are used, the interlaced iteration is economical to use.

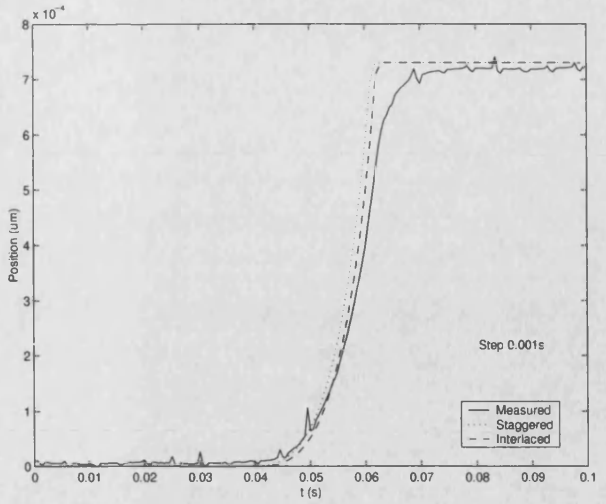


Fig. 7. Measured and simulated position with $\Delta t = 10^{-3}$ s.

TABLE III
Total number of linear equation solutions.

Δt	Staggered	Interlaced	Percent increase
10^{-2} s	56	68	19
10^{-3} s	478	572	21

V. Convergence analysis

A convergence analysis of the block Gauss–Seidel iteration has been carried out by [3]. From one iteration to the next, the error $\mathbf{e}$ is amplified according to

$$\mathbf{e}^{i+1} = -(\mathbf{J}_L + \mathbf{J}_D)^{-1}\mathbf{J}_U\mathbf{e}^i, \qquad (5)$$

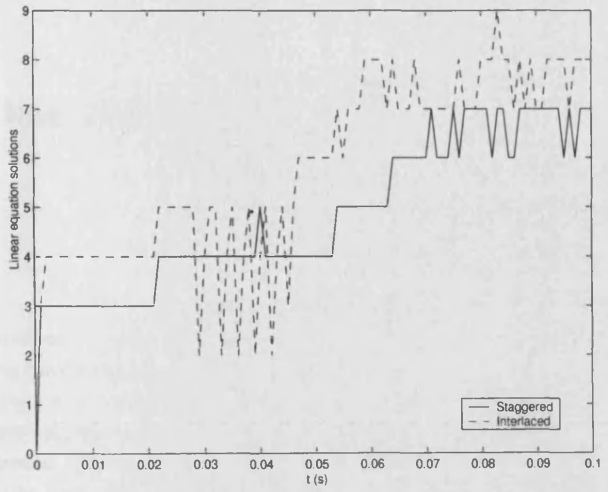


Fig. 8. Number of linear equation solutions (electrical part) with $\Delta t = 10^{-3}$ s.

where $\mathbf{J}$ is the Jacobian of the system equations (1)-(2), and L, D and U stand for strictly lower, diagonal and strictly upper matrices. Observe that the convergence rate is not influenced by the choice of nested nonlinear solver. The interlaced iteration has a convergence rate given by

$$\mathbf{e}^{i+1} = (\mathbf{I} - \mathbf{H}_D\mathbf{J}_L)^{-1}\left[\mathbf{I} + \mathbf{H}_D\left(\mathbf{J}_D + \mathbf{J}_U\right)\right]\mathbf{e}^i, \qquad (6)$$

where $\mathbf{H}_D$ is the nonlinear iteration matrix [3]. If a Newton–Raphson method is used, then $\mathbf{H}_D = -\mathbf{J}_D^{-1}$ and (6) leads to

$$\mathbf{e}^{i+1} = \left(\mathbf{I} + \mathbf{J}_D^{-1}\mathbf{J}_L\right)^{-1}\left(-\mathbf{J}_D^{-1}\mathbf{J}_U\right)\mathbf{e}^i. \qquad (7)$$

Pre-multiplying both sides by $\left(\mathbf{I} + \mathbf{J}_D^{-1}\mathbf{J}_L\right)$ and then by $\mathbf{J}_D$:

$$\left(\mathbf{J}_D + \mathbf{J}_L\right)\mathbf{e}^{i+1} = -\mathbf{J}_U\mathbf{e}^i, \qquad (8)$$

which can be seen to be the same as (5). Thus, the interlaced iteration has the same convergence characteristics as the block Gauss–Seidel when a Newton–Raphson method is used. However, each block Gauss–Seidel iteration requires several linear equation solutions for each domain, while each interlaced iteration requires only one.

The convergence rate of a global Newton–Raphson method is given by

$$\mathbf{e}^{i+1} = (\mathbf{I} + \mathbf{H}\mathbf{J})\,\mathbf{e}^i. \qquad (9)$$

In the special case of a global Newton–Raphson process, the amplification is zero. According to this, the error is eliminated in a single iteration. In practice, several iterations are required because the first order approximation implied in (9) is not exact in the general case.

The block and interlaced iterations have an amplification matrix which is not zero even if the first order approximation is exact. This has several consequences. First, the block and interlaced methods will require more iterations to converge compared to a global nonlinear iteration. Second, they may fail to converge, this depending on the nature of the coupled system. Third, if the interlaced iteration fails to converge, the block iteration will also fail.

VI. Conclusions

A simple procedure to achieve economical solutions of coupled problems has been outlined. The method is easy to implement and provides strong coupling between domains. It was implemented for an equilibrium–type electromechanical problem, and found to converge. An analysis of the convergence rate shows that the procedure is equivalent to the usual Gauss–Seidel block iteration, but the number of linear equation solutions is reduced. The same strategy applied to a dynamic problem leads to a consistent solution (strong coupling) at each time step, allowing a larger time step to be used for a given accuracy. The added computational cost is about 20% compared to the staggered solution.

Acknowledgment

The authors thank Dr. P. C. Coles for providing the experimental results.

References

[1] P. Eustache, G. Meunier, and J. L. Coulomb, "Finite element toolbox for generic coupling (magnetic, thermal,etc.)," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1461–1464, May 1996.

[2] M. Cervera, R. Codina, and M. Galindo, "On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems," *Engineering Computations*, vol. 13, no. 6, pp. 4–30, 1996.

[3] I. Doltsinis, "Solution of coupled systems by distinct operators," *Engineering Computations*, vol. 14, no. 8, pp. 829–868, 1997.

[4] C. Bailey, G. A. Taylor, M. Cross, and P. Chow, "Discretisation procedures for multi-physics phenomena," *Journal of Computational and Applied Mathematics*, vol. 103, no. 1, pp. 3–17, 1999.

[5] Z. Ren and A. Razek, "A strong coupled model for analysing the dynamic behaviour of non-linear electromechanical systems," *IEEE Transactions on Magnetics*, vol. 30, no. 5, pp. 3252–3255, September 1994.

[6] Z. Ren, B. Ionescu, M. Besbes, and A. Razek, "Calculation of mechanical deformation of magnetic materials in electromagnetic devices," *IEEE Transactions on Magnetics*, vol. 31, no. 3, pp. 1873–1876, May 1995.

[7] Z. Johan, T. J. R. Hughes, and F. Shakib, "A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids," *Computer Methods in Applied Mechanics and Engineering*, , no. 87, pp. 281–304, 1991.

[8] J. H. Prevost, "Partitioned solution procedure for simultaneous integration of coupled-field problems," *Commun.Numer.Meth.Engng*, vol. 13, pp. 239–247, 1997.

[9] N. R. Aluru and J. White, "A multilevel newton method for mixed-energy domain simulation of mems," *Journal of Microelectromechanical Systems*, vol. 8, no. 3, pp. 299–307, September 1999.

[10] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method Vol.II: Solid and Fluid Mechanics, Dynamics and Non-linearity*, McGraw Hill, London, 1989, Fourth Edition.

[11] H. C. Lai, P. C. Coles, D. Rodger, and P. J. Leonard, "Transient analysis of an electromagnetic actuator using an overlapping finite element scheme," *IEEE Transactions on Magnetics*, vol. 36, no. 4, pp. 1462–1467, July 2000.

Time Transient Dynamic Analysis of a Rotational Test Rig Benchmark Problem

H.C. Lai, P.C. Coles, E.Melgoza, D. Rodger, P.J. Leonard and R.J.Hill-Cottingham

***Abstract*—The transient characteristics of an experimental test rig consisting of simple shaped solid iron stationary stator and rotating rotor under a step voltage input is measured. A 3D finite element scheme which takes into account the non-linear magnetic properties, the eddy current effects and the dynamic motion of the rotor was also used to simulate the response of the rig under the same experimental set up. The measurement was used as a benchmark against which the simulation results were compared.**

***Index Terms*—Finite Element Methods, Transient Analysis, Coupled Problems, Eddy Current**

I. Introduction

There are electromagnetic devices which relies on the eddy current generated in the solid moving conductors part contained within to function properly. Examples of these include induction motors and eddy current braking systems. When finite elements is used to analyse such devices, it is crucial that the eddy current generated in, the non-linear magnetic properties and the motion of the various parts are accurately modelled. If the moving conductor has a constant conductivity, a constant cross-section normal to the direction of velocity and moving in a constant velocity, then the Minkowski transformation scheme can be used. However, if the aim is to model, for example, the start up transient of a squirrel cage induction motor, then a proper time stepping scheme which can handle all the requirements mentioned above must be used. This type of transient finite elements models is very demanding numerically and is expensive to solve. It is extremely valuable that results obtained using existing and new finite element formulations can be verified against benchmark experimental results.

This paper describes the modelling of the transient dynamic behaviour of an experimental test rig under step voltage forced input using a nonlinear 3D finite element time stepping scheme. The dynamic movement of the rotating part is handled by using the Lagrange sliding interface technique [1]. Fig. 1 shows an image of the test rig used. The results obtained from the finite element analysis are compared with experimental measurements.

II. Finite Element Model

The total and reduced magnetic scalar potentials are used in non-conducting regions [2]. The basic method has been extended to allow voltage forced conditions [3] and to produce

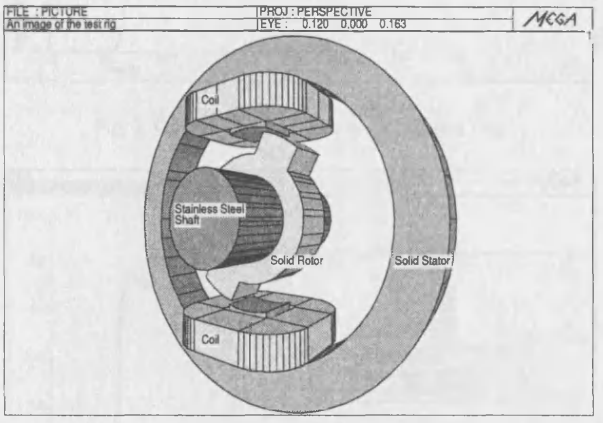


Fig. 1. An image of the test rig

cuts for handling multiply connected problems [4].

In conducting regions, fields can be modelled using $\mathbf{A}$, the magnetic vector potential, and V, the electric scalar potential, we have

$$\nabla\times\left(\frac{1}{\mu}\nabla\times\times\mathbf{A}\right)+\sigma\left(\frac{\partial\mathbf{A}}{\partial t}+\nabla V\right)=0 \qquad (1)$$

$$\nabla\cdot\left(\sigma\frac{\partial\mathbf{A}}{\partial t}+\sigma\nabla V\right)=0 \qquad (2)$$

When the conducting region is moving at a velocity, equation (1) can be written as [5]

$$\nabla\times\left(\frac{1}{\mu}\nabla\times\mathbf{A}\right)=-\sigma\left[\frac{\partial\mathbf{A}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{A}+\mathbf{A}\times\boldsymbol{\omega}\right] \qquad (3)$$

where $\boldsymbol{\omega}$ is the angular velocity.

The first two terms on the right hand side of equation(3) are taken into account if the mesh representing the moving conducting part is allowed to move relative to the mesh representing the stationary part. The third term is required if the movement is a rotational one. These can be solved using a Galerkin finite element scheme and the nonlinear system can be tackled using a Newton Raphson scheme.

III. Rotor Movement

To simulate the dynamic movement of the moving part, the model was solved as a time transient problem. The Lagrange sliding interface scheme used allows the stationary and moving mesh to rotate relative to each other so that the displacement of

the latter over time are modelled. The following simple equation was used to estimate the displacement of the moving part during each time step.

$$I\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + T_f = T_{em} \tag{4}$$

where I is the inertia of the moving part, θ is the angular displacement, T_f is the external force and T_{em} is the electromagnetic torque on the moving part. T_f, in this case, is the frictional torque the value of which is dependent on the rotating velocity. T_{em} is obtained form the electromagnetic solution. The rotating mesh is moved by the displacement calculated by (4) before the start of the next time step.

A second scheme, which we will call the 'Interlaced Nonlinear Iteration', was also used to solve the combined electromagnetic-mechanical system. In this scheme, the uncoupled systems are solved alternatively within a Newton solver until both systems have converged. These steps can be specified with the pseudo-code:

I- Set $\mathbf{x}_1 = \mathbf{x}_1^0, \mathbf{x}_2 = \mathbf{x}_2^0$.

II- Loop on i until convergence.

1) Get $\mathbf{x}_1^{i+1} = \mathbf{x}_1^i + \mathbf{J}_{11}\mathbf{f}_1(\mathbf{x}_1^i, \mathbf{x}_2^i,)$.

2) Get $\mathbf{x}_2^{i+1} = \mathbf{x}_2^i + \mathbf{J}_{22}\mathbf{f}_2(\mathbf{x}_1^{i+1}, \mathbf{x}_2^i)$.

Here, x_1 is the vector of magnetic variables, x_2 is the vector of mechanical variables, and $\mathbf{J}_{11}$ and $\mathbf{J}_{22}$ are the domain Jacobians.

IV. Test rig and results

The test rig (Fig. 1) used in the TEAM Workshop Problem 24 [6] was adopted to use in the experiment. It resembles a 2-2 switched reluctance motor and consists of a rotor pole mounted on a non-magnetic stainless steel shaft which is free to move inside a non-moving stator. Both the rotor and stator are made of solid steel so that nonlinear and eddy current effects are significant everywhere. The detail dimensions of the rig have been published elsewhere [6]. The starting and low speed frictional torque values on the moving rotor part was measured and used in the simulation.

The corner of the rotor pole was initially aligned to the corner of the stator pole and a step voltage of $5.97V$ was applied to the series coils mounted on the two stator poles. The rotor was allowed to move under the influence of the electromagnetic torque. The transient coil current and the displacement of the rotor was measured.

Fig. 2 shows the measured and calculated displacement of the rotor against time. Fig. 2 compares the measured and calculated current in the source coils against time.

V. Conclusion

The transient measurements on an experiment solid steel test rig containing a rotating rotor are presented. These are compared with simulation results obtained using a 3D finite element nonlinear time stepping scheme which can simulate the

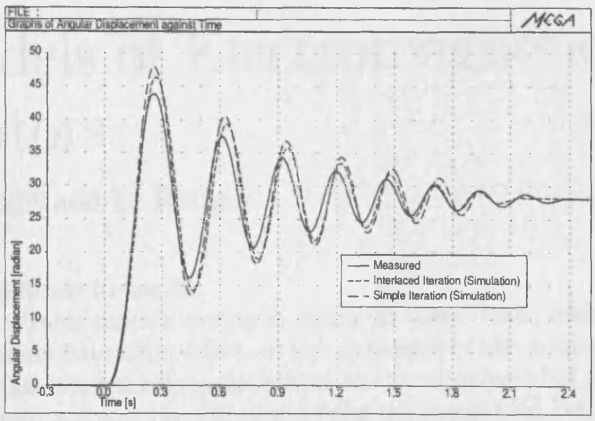


Fig. 2. Graphs of rotor displacement against time

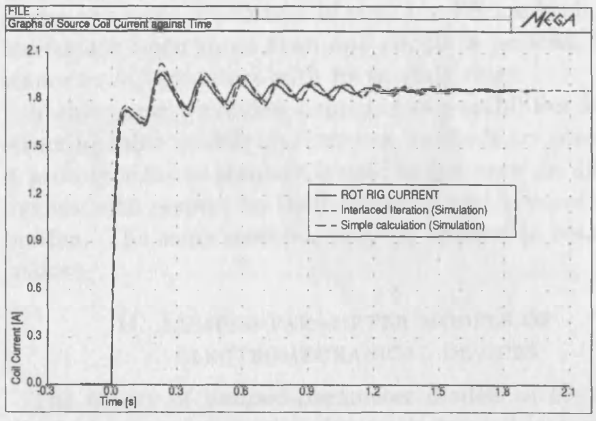


Fig. 3. Graphs of coil current against time

dynamic rotation of the rotor occurring during the experiment. Comparisons between the measured and simulated results, although reasonable, highlight the valuable contribution a realistic experimental set up can offer to the verification of new and existing complex numerically models.

References

[1] H.C.Lai, D.Rodger, and P.J.Leonard. "Coupling meshes in 3D problems involving movement". *IEEE Trans. Magn.*, 28(2):1732–1734, March 1992.

[2] J.Simkin and C.W.Trowbridge. "On the use of the total scalar potential in the numerical solution of three dimensional magnetostatic problems". *IJNME*, 14:423–440, 1979.

[3] P.J.Leonard and D.Rodger. "Modelling voltage forced coils using the reduced scalar potential method". *IEEE Trans. Magn.*, 28(2):1615–1617, March 1992.

[4] P.J.Leonard, H.C.Lai, R.J.Hill-Cottingham, and D.Rodger. "Automatic implementation of cuts in multiply connected magnetic scalar regions for 3D eddy current models". *IEEE Trans. Magn.*, 29(2):1368–1371, 1993.

[5] N.Allen, H.C.Lai, D.Rodger, and P.J.Leonard. "On the validity of two $A\psi$ finite element formulations for modelling eddy current problems with velocity". *IEEE Trans. Magn.*, 34(5):2535–2538, September 1998.

[6] N.Allen and D.Rodger. Nonlinear time-transient rotational test rig. *Proc. of the TEAM Workshop in the Sixth Round - Okayama, Japan*, pages 57–60, March 1996.

Comparison of Table Models of Electromagnetic Actuators

E. Melgoza, *Member, IEEE* and D. Rodger

Abstract— **A number of possibilities for constructing table-based lumped-parameter models of linear or rotary actuators are compared. Two of them have not been published previously. The accuracy of the models is evaluated by comparing simulated results against experimental measurements taken from a prototype actuator. The computational cost of the models is also discussed.**

Keywords— **Actuators, Modeling, Finite Element Method, Lumped Parameters**

I. Introduction

ACCURATE analysis of electromagnetic actuators is required in demanding applications such as fuel injection control [1]. The response of the device is determined by mechanical and electrical variables behaving nonlinearly, to the extent that a field model is often necessary. The finite element method or some other discretization procedure may be used to that end [1], [2]. However, the resulting model comprises a large number of equations, making it expensive to use when several variations need to be considered. Additionally, such models are difficult to integrate in system-level simulators, where the overall performance of the actuator–controller system is being analyzed or designed, or when the actuator forms part of a larger system with other components to be modeled as well.

When dealing with the overall performance of the device-controller system, a small model is convenient to have in order to reduce the simulation time [3]. For the special but important case of single coil actuators, linear or rotary, in which there is only one electrical and one mechanical degree of freedom, table methods have been used, combining the accuracy of the field methods with the speed of lumped parameter models [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. The idea is to use a field method to determine the lumped parameters of a reduced model. Since only a finite number of points can be calculated, the points are arranged in a table and an interpolation method has to be applied or an analytical function fitted to determine the dynamics of the system. The tables are constructed by an off-line finite element (FE) simulation (or they could be measured) for a grid of position and current (or position and flux linkage) values. The resulting model captures the nonlinear behavior of the device in an economical manner, but due to the fact that a set of tables has to be calculated, this approach is restricted to the simplest devices, namely those with only a few coils. Otherwise the model becomes difficult to handle.

Table models are most useful in cases where eddy currents have little effect on the dynamics of the actuator. If this is not the case, the model can be supplemented to take into account the diffusion of the magnetic field [14]. One related approach consists in linking an on-line FE calculation of the lumped parameters of the device [15]. This shares the same drawbacks of the only-FE method, but is convenient when more than one circuit is present, for instance in AC actuators with its shading rings.

In this paper we review a number of possibilities for constructing table models and two new methods are presented. A prototype linear actuator is used to compare the different models with respect to their accuracy and computational burden. The same methods may be applied to rotary actuators.

II. Lumped-parameter models of electromechanical devices

The theory of lumped-parameter models of electromechanical energy conversion devices is treated in a number of texts [16]. If the state variables are the flux linkage and position (λ, x), then the current i and force f_e are obtained by differentiating the magnetic energy W_m:

$$i = \frac{\partial W_m}{\partial \lambda}, \tag{1}$$

$$f_e = -\frac{\partial W_m}{\partial x}. \tag{2}$$

On the other hand, if the state variables are the current and position (i, x) the magnetic coenergy W'_m is used:

$$\lambda = \frac{\partial W'_m}{\partial i} \tag{3}$$

$$f_e = \frac{\partial W'_m}{\partial x} \tag{4}$$

The energy and coenergy are related by

$$W_m + W'_m = \lambda i \tag{5}$$

III. State variables

Three state variables are required for simulating the actuator. The mechanical state equations are always the same, given by

$$\dot{v} = \frac{1}{M}[F_e - Bv - K(x - x_0)], \tag{6}$$

$$\dot{x} = v, \tag{7}$$

D. Rodger is with the Applied Electromagnetics Research Centre, University of Bath. BATH, BA2 7AY, UK.

E.Melgoza is with Instituto Tecnologico de Morelia, Morelia, Mich. 58120 Mexico

This work was supported in part by CONACYT and SEP.

where v is the velocity, x the position, M the mass, B is the damping coefficient, K is the spring constant and x_0 the spring neutral position.

The expressions for the electrical state variable and electromagnetic force F_e depend on the method used. Four possibilities are discussed now.

IV. Table methods

Equations (1)-(2) and (3)-(4) are the starting point for constructing a model based on tables. Several approaches are possible, but in practice the selection of a particular method will be determined by the results available from the FE computation. For instance, if the finite element program is capable of calculating the force, then a function $F_e(i,x)$ can be constructed. But if the force is not available then another method must be used. FE programs calculate the magnetic vector potential from a given current density distribution; this means that the (i,x) formulation is a natural way of developing the lumped-parameter model. However, (λ,x) formulations are also possible, as we will see. A plane 2D, axisymmetrical or 3D FE model may be used, depending on the particular problem at hand.

A. $(\mathbf{A}, F_e)$ *Model*

As mentioned above, the FE program most likely will require the specification of the current density, from which the magnetic vector potential $\mathbf{A}$ will be calculated. If the force is available from the FE program, then a series of calculations for a grid of current values and positions will provide us with a pair of tables

$$\mathbf{A} = \mathbf{A}(i,x) \tag{8}$$

$$F = F(i,x) \tag{9}$$

The flux linkage can be computed from (8) by using

$$\lambda = N \oint_L \nabla \times \mathbf{B} \cdot d\mathbf{L} = N \oint_L \mathbf{A} \cdot d\mathbf{L}, \tag{10}$$

where N is the number of turns in the coil. As a result a table of flux linkage $\lambda(i,x)$ will be available. The electrical circuit equation is

$$V = Ri + \frac{d\lambda}{dt} = Ri + \frac{\partial\lambda}{\partial i}\frac{di}{dt} + v\frac{\partial\lambda}{\partial x}, \tag{11}$$

where V is the applied voltage and R the electrical resistance. From (11), the electrical state variable is the current, given by

$$\frac{di}{dt} = \frac{1}{\frac{\partial\lambda}{\partial i}}\left(V - Ri - v\frac{\partial\lambda}{\partial x}\right). \tag{12}$$

This method has been used by [4], [8], [9], [11], [12]. The coil will contain several nodes of the FE model, so that the location of the sampling point from which $\mathbf{A}(i,x)$ is taken is not unique (the values differ little but are not uniform over the region of the coil). Some kind of average process is desirable or, alternatively, the value at a given node placed for instance at the center of the coil may be used. The later approach generally gives good results.

B. (W_m, W'_m) *Model*

If both the energy and coenergy can be calculated, then the flux linkage is obtained from (5) when the current is specified; the electrical state variable is the current and is again given by (12). On the other hand, the force is now calculated by using (2). The flux linkage so calculated is an averaged value since it is calculated from integral expressions (the energy and coenergy in this case) . This method has not been reported before.

C. (W'_m) *Model*

It is possible to construct a model from only the knowledge of the coenergy. In this case, the flux linkage has to be computed from (3). Substituting in the circuit equation (11):

$$V = Ri + \frac{d\lambda}{dt} = Ri + \frac{\partial^2 W'_m}{\partial i^2}\frac{di}{dt} + \frac{\partial^2 W'_m}{\partial i \partial \lambda}v. \tag{13}$$

The state equation is then

$$\frac{di}{dt} = \frac{1}{\frac{\partial^2 W'_m}{\partial i^2}}\left(V - Ri - v\frac{\partial^2 W'_m}{\partial i \partial \lambda}\right) \tag{14}$$

The force is calculated directly by using (4). This approach has been proposed by [13].

D. $(\mathbf{A})$ *Model*

The three preceding models use the (i,x) formulation of the energy conversion process. It is also possible to use the (λ,x) formulation, but some modifications to the table data from the FE runs are needed. If the magnetic vector potential $\mathbf{A}(i,x)$ is available, then it is possible to find an inverse function $i(\lambda,x)$. This can be seen in Fig,1, where the calculated flux linkage for a prototype actuator has been plotted against the current. The flux linkage-current relation is linear for large air gaps, but it gradually transforms into a curve following the nonlinear magnetization characteristic of the material for small air gaps. In any case, the relation is one to one and thus an inverse function $i(\lambda,x)$ can be calculated.

Once the inverse function $i(\lambda,x)$ has been found, the energy can be calculated by using

$$W_m = \int i d\lambda, \tag{15}$$

according to (1).

The electric state variable is λ in this case, the circuit equation being

$$V = Ri + \frac{d\lambda}{dt} \tag{16}$$

and the state equation

$$\frac{d\lambda}{dt} = V - Ri. \tag{17}$$

In (17), the current i is evaluated from the previously calculated function $i(\lambda,x)$, or it may be approximated from (1), in which case a single table is required since the force

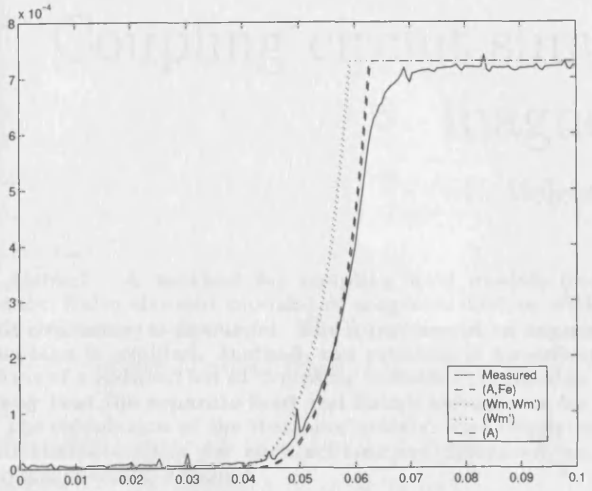


Fig. 3. Transient position during closing.

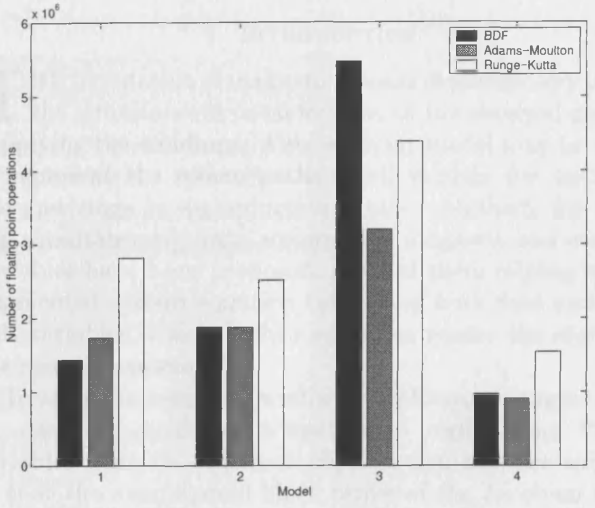


Fig. 4. Number of floating point operations for the distinct models: 1-$(\mathbf{A}, F_e)$, 2-(W_m, W'_m), 3-(W'_m), 4-$(\mathbf{A})$.

then the circuit equation (11) leads to

$$\frac{di}{dt} = \frac{1}{L + i\frac{\partial L}{\partial i}} \left[V - \left(R + v\frac{\partial L}{\partial x} \right) i \right]. \qquad (19)$$

When used together with a force table, this method is equivalent to $(\mathbf{A}, F)$. Very often, the $i\frac{\partial L}{\partial i}$ term in the denominator of (19) is omitted, but this is only valid for large air gaps, where the inductance may be regarded as a function only of the position, $L = L(x)$.

VIII. Conclusions

Several alternatives for constructing table models of electromagnetic energy conversion devices with one electrical and one mechanical degrees of freedom have been discussed, two of them thought to be new. A closing transient was simulated and the results compare well against measurements. The computational cost was evaluated, from where it was found that the $(\mathbf{A})$ model requires the least number of floating point operations. All the notation and discussion has assumed a one-coil device, but the methods are still applicable to multi-coil cases, except (W_m, W'_m); the relevant equations assume a matrix form instead of the scalar form presented.

References

[1] P. S. Sangha and D. Rodger, "Design analysis of voltage fed axisymmetric actuators," *IEEE Transactions on Magnetics*, vol. 30, no. 5, pp. 3240–3243, September 1994.

[2] Y. Kawase and Y. Ohdachi, "Dynamic analysis of automotive solenoid valve using finite element method," *IEEE Transactions on Magnetics*, vol. 27, no. 5, pp. 3939–3942, September 1991.

[3] I. R. Smith, J. G. Kettleborough, and Y. Zhang, "Simplified modelling and dynamic analysis of a laws' relay actuator," *Mechatronics*, vol. 9, no. 5, pp. 463–475, 1999.

[4] L. Bompa, P. Schueller, and J. C. Sabonnadiere, "Analysis and synthesis of an electromagnet used for circuit breaker operation," *IEEE Transactions on Magnetics*, vol. 21, no. 6, pp. 2464–2467, November 1985.

[5] Y. Weili, R. Liyun, Y. Qinxin, L. Yongchao, S. Xueqin, and W. Jingguo, "Finite element dynamic analysis for the electromagnet of AC contactor," *IEEE Transactions on Magnetics*, vol. 27, pp. 4133–4135, September 1991.

[6] L. Erping and P. M. McEwan, "analysis of a circuit breaker solenoid actuator system using the decoupled cad-fe-integral technique," *IEEE Transactions on Magnetics*, vol. 28, no. 2, pp. 1279–1282, March 1992.

[7] S. Wang, T. Miyano, and M. Hubbard, "Electromagnetic field analysis and dynamic simulation of a two-valve solenoid actuator," *IEEE Transactions on Magnetics*, vol. 29, no. 2, pp. 1741–1746, March 1993.

[8] Z. Ren and A. Razek, "Modelling of dynamical behaviour of electro-magneto-mechanical coupled systems," in *Proceedings of the IEE Second International Conference on Computation in Electromagnetics*, April 1994, Conference Publication Number: 384.

[9] K. Srairi, M. Feliachi, and Z. Ren, "Elecromagnetic actuator behaviour analysis using finite element and parametrisation methods," *IEEE Transactions on Magnetics*, vol. 31, no. 6, pp. 3497–3499, November 1995.

[10] N. Sadowski, R. Carlson, A. M. Beckert, and J. P. A. Bastos, "Dynamic modeling of a newly designed linear actuator using 3D edge element analysis," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1633–1636, May 1996.

[11] T. E. McDermott, P. Zhou, J. Gilmore, and Z. Cendes, "Electromechanical system simulation with models generated from finite element solutions," *IEEE Transactions on Magnetics*, vol. 33, pp. 1682–1685, March 1997.

[12] A. Canova, M. Otella, and D. Rodger, "A coupled field-circuit approach to 3D FEM analysis of electromechanical devices," in *Ninth International Conference on Electrical Machines and Drives*. 1999, pp. 71–75, IEE, Conference Publication No.468.

[13] R. Gollee, T. Roschke, and G. Gerlach, "A finite element method-based dynamic analysis of a long-stroke linear actuator," *Journal of Magnetism and Magnetic Materials*, vol. 197, pp. 943–945, 1999.

[14] J. R. Brauer and Q. M. Chen, "Alternative dynamic electromechanical models of magnetic actuators containing eddy currents," in *Compumag99, 12th Conference on the Computation of Electromagnetic Fields*, Sapporo, Japan, October 1999, pp. 40–41, Conference Record.

[15] N. A. Demerdash and T. W. Nehl, "Electric machinery parameters and torques by current and energy perturbations from field computations - Part I: Theory and formulation," *IEEE Transactions on Energy Conversion*, vol. 14, no. 4, pp. 1507–1513, December 1999.

[16] D. C. White and H. H. Woodson, *Electromechanical Energy Conversion*, John Wiley & Sons, New York, 1959.

[17] H. C. Lai, P. C. Coles, D. Rodger, and P. J. Leonard, "Transient analysis of an electromagnetic actuator using an overlapping finite element scheme," *IEEE Transactions on Magnetics*, vol. 36, no. 4, pp. 1462–1467, July 2000.

Coupling circuit simulators and field models of magnetic devices

E. Melgoza, *Member, IEEE*

Abstract— **A method for coupling field models (for instance, finite element models) of magnetic devices with circuit simulators is discussed. The formation of an augmented Jacobian is avoided. Instead, the problem is formulated in terms of a reduced set of 'coupling variables', defined in such a way that the separate field and circuit solvers can be used in the calculation of the iteration matrix. The requirements and characteristics for such scheme are discussed, and its implementation detailed.**

Keywords— **Finite Element Method, Lumped Parameters, Coupled Problems**

I. Introduction

THE simulation of magnetic devices demands very often the simultaneous consideration of the external circuit supplying the windings. Also, a circuit model may be used to represent the return paths in 2D models, for instance the end rings in an induction motor. Methods for taking simultaneously into account the magnetic and circuit variables have been proposed, most of them relying in an augmented system equation containing both field and circuit variables, which is then solved to render the state of the coupled system.

To solve the resulting nonlinear problem, an augmented Jacobian is formed, which has several implications. First, variables from two distinct physical domains are mixed, so that the nondiagonal block terms of the Jacobian may be difficult to calculate. A special formulation is often required if sparsity and simmetry of the augmented matrix is to be conserved. Second, existing computer codes dealing with circuit simulation are not used. Depending on the nature of the problem at hand, a power system-oriented or an electronic devices-oriented simulator may be appropriate.

A coupled solution scheme which uses specialized circuit simulators while still showing fast convergence is desirable. Recently, one such scheme has been proposed, using the concept of multiport networks to embed the finite element model into the circuit [1]. The finite element model can be seen to have been 'absorbed' by the circuit model after linearization.

In this paper, we propose one scheme for coupling a finite element model with a circuit simulator. In our approach, both domains are treated separately, allowing the use of existing codes albeit after suitable modification as discussed later. Communication between domains is achieved through a subset of variables whose number is small compared to the number of magnetic variables.

The author was with the Applied Electromagnetics Research Center, University of Bath. He is now with Instituto Tecnologico de Morelia, Morelia, Mich. 58120, Mexico

This work has been supported by Conacyt and SEP.

From a more general point of view, this paper addresses the question of the coupling between separate transient solvers in the solution of coupled problems. Separate solvers may offer some advantages, among them the use of specialized codes [2].

II. Coupling variables

To understand the concept of coupling variables, suppose we attempt to solve the field-circuit problem by a Gauss–Seidel iteration. The sequence could be, for each time step:

1. Set the currents in the windings.
2. Solve the magnetic problem to get the magnetic vector potentials and from them the emf's (electromotive forces) in the windings.
3. Solve the circuit equation, regarding the emf in each winding as a voltage source to obtain the current in the windings.
4. Repeat until convergence is achieved.

This iterative scheme may be slow or even fail to converge. The important point to observe is that there is a definite set of variables communicating between 'modules': the currents are input to the field model and the result is a set of winding voltages. These voltages are input to the circuit model, and the output is a set of corrected currents. Thus, the coupled problem can be stated in terms of this 'communication' or 'coupling' variables. What we seek is a set of voltages $\mathbf{e}$ and currents $\mathbf{i}$ satisfying the set of nonlinear equations

$$\begin{aligned} \mathbf{e} - \mathbf{e}_f(\mathbf{i}) &= \mathbf{0} \\ \mathbf{i} - \mathbf{i}_c(\mathbf{e}) &= \mathbf{0}. \end{aligned} \tag{1}$$

In these equations, $\mathbf{e}_f$ is a transformation rule $\mathbf{i} \rightarrow \mathbf{e}$ and the subscript f is used to denote that this transformation is achieved by solving the magnetic field equations. Similarly, $\mathbf{i}_c$ performs the transformation $\mathbf{e} \rightarrow \mathbf{i}$, c indicating that this transformation is achieved by solving the circuit equation [3].

A Newton iteration can be used instead of the Gauss-Seidel scheme, in which case the iteration equation is, from (1):

$$\begin{bmatrix} \mathbf{I} & -\frac{\partial \mathbf{e}_f}{\partial \mathbf{i}} \\ -\frac{\partial \mathbf{i}_c}{\partial \mathbf{e}} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{e} \\ \Delta \mathbf{i} \end{bmatrix} = \begin{bmatrix} -\mathbf{e} + \mathbf{e}_f(\mathbf{i}) \\ -\mathbf{i} + \mathbf{i}_c(\mathbf{e}) \end{bmatrix}. \tag{2}$$

This equation has dimension $2w$, where w is the number of windings. It is worth noting that the details of the scheme are the same for other cases of coupling between field and lumped parameter models. For example,

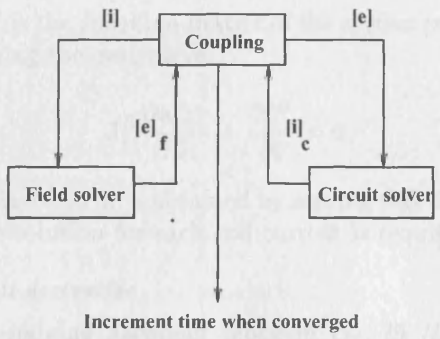


Fig. 1. Coupling scheme for one time step.

a thermal circuit may be used to model the temperature dependence.

III. Time stepping

The scheme sketched above has been used to solve static coupled problems using separate solvers treated as black boxes [3]. Here we will use it for solving transient problems. We assume that separate transient solvers exist for the field and circuit domains; each solver is started and runs separately, but we require that at each time step the nonlinear equation (1) is satisfied. Thus, the control program must ensure that the appropiate communication takes place; the solvers are responsible for calculating the next state of their respective domains and also return the value of the coupling variables and their derivatives, which will be used by the control program to solve (2) The sequence is illustrated in Fig. 1.

IV. Implementation

To make procedures clear, consider a two-dimensional model in cartesian coordinates, for which the z-component of the magnetic vector potential, A_z, obeys the well known relation

$$\nabla \cdot \nu \nabla A_z - \sigma \frac{\partial A_z}{\partial t} + J_w = 0, \tag{3}$$

where ν is the reluctivity, σ is the conductivity, and J_w is the current density in the winding, given by

$$J_w = \frac{N i_w}{S_w}. \tag{4}$$

Here, S_w is the area of the winding. On the other hand, the emf is given by

$$e_w = \frac{NL}{S_w} \int \frac{dA_z}{dt} dS_w. \tag{5}$$

Equations (3)-(5) constitute the transformation rule $\mathbf{e}_f(\mathbf{i})$ in (1). The $\partial \mathbf{e}_f / \partial \mathbf{i}$ term of the Jacobian matrix in (2) is given by

$$\frac{\partial \mathbf{e}_f}{\partial \mathbf{i}} = \frac{\partial \mathbf{e}_f}{\partial \mathbf{a}} \frac{\partial \mathbf{a}}{\partial \mathbf{i}}, \tag{6}$$

where $\mathbf{a}$ is the vector of magnetic vector potentials.

A. EMF derivative

The first partial derivative in the right hand side of (6), $\partial \mathbf{e}_f / \partial \mathbf{a}$, can be found from (5) after the time derivative dA_z/dt has been approximated by a suitable discrete formula. For instance, if we use the first order approximation

$$\frac{1}{h}(A_z - A_z^n), \tag{7}$$

where h is the time step size and A_z^n is the past state (and therefore a constant), then the emf is given by

$$e_f = \frac{NL}{hS_w} \left(\int_S A_z dS_w - \int_S A_z^n dS_w \right). \tag{8}$$

Since the second term in (8) is constant, we concentrate our attention on the first term only, which results from adding the contributions of the all elements in the model:

$$\frac{NL}{hS_w} \int_S A_z dS_w = \frac{NL}{hS_w} \sum_E \int_S A_z dS_E. \tag{9}$$

Replacing the expression for A_z in terms of the shape functions N_i and the nodal values A_i

$$\frac{NL}{hS_w} \int_S A_z dS_w = \frac{NL}{hS_w} \sum_E \sum_i \int_S N_i A_i dS_E. \tag{10}$$

The emf derivative is therefore the vector with entries:

$$\frac{\partial e_f}{\partial A_i} = \frac{NL}{hS_w} \sum_E \int_S N_i dS_E. \tag{11}$$

Examination of (11) reveals that it has the same form as the source term of the finite element discretization of (3), so that its calculation and assembly are made by using available routines.

B. MVP derivative

The evaluation of the other derivative, $\partial \mathbf{a} / \partial \mathbf{i}$, is more involved. First, discretization of (3) in the spatial variables x and y, for instance by using the finite element method, leads to the expression

$$\mathbf{C}\dot{\mathbf{a}} + \mathbf{K}(\mathbf{a})\mathbf{a} + \mathbf{f}(\mathbf{i}) = \mathbf{0}. \tag{12}$$

Then we use some time stepping formula to discretize in time. If we choose the backward Euler formula we have

$$\left(\frac{1}{h}\mathbf{C}_{n+1} + \mathbf{K}_{n+1} \right) \mathbf{a}_{n+1} + \left(-\frac{1}{h}\mathbf{C}_{n+1}\mathbf{a}_n + \mathbf{f}_{n+1} \right) = \mathbf{0}, \tag{13}$$

where $h = t_{n+1} - t_n$ is the time step. This is a nonlinear algebraic equation that can be written in the equivalent form

$$\mathbf{K}^D(\mathbf{a})\mathbf{a} + \mathbf{f}^D(\mathbf{i}) = \mathbf{0}, \tag{14}$$

where the superscript D indicates full space and time discretization. The corresponding Newton iteration formula is

$$\mathbf{J}_i^D(\mathbf{a}_{i+1} - \mathbf{a}_i) + \mathbf{K}_i^D \mathbf{a}_i + \mathbf{f}_i^D = \mathbf{0}, \tag{15}$$

where $\mathbf{J}^D$ is the Jacobian matrix of the system of equations (14). Taking the derivative:

$$\mathbf{J}_i^D \frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{i}} + \frac{\partial \mathbf{f}_i^D}{\partial \mathbf{i}} = \mathbf{0}, \tag{16}$$

and the $\partial \mathbf{a}/\partial \mathbf{i}$ term is obtained by solving this linear equation. One solution for each coil current is required.

C. Circuit derivative

The remaining Jacobian block in (2), $\partial \mathbf{i}_c/\partial \mathbf{e}$, can be found by manipulating the circuit equations in a similar way. This could be based in the more developed theory of circuit sensitivity analysis, from which time domain sensitivity expressions are available [4]. Similar expressions for finite element analysis are unknown to the author; time harmonic expressions for linear systems have been given by [5].

D. Interlacing

Three Newton processes are involved here. One is the Newton iteration for the coupled problem, given by (2), which contains the other two Newton iterations for the magnetic field, given by (15), and for the circuit. If full solution of the inner Newton iterations is performed, then the solution of the whole transient problem is costly. On the other hand, an interlaced Newton process can be attempted, resulting in only one iteration of the inner nonlinear problems for each iteration of the outer nonlinear problem [6].

V. Example

To test the scheme, an inductor modelled by the finite element model is energized by a DC source with a series resistance. The resulting emf and current in the inductor are shown in Figs. 2 and f:circuit1-i and compared against the solution given by a conventional solver using an augmented system matrix. When the problem is solved with interlaced iteration, we expect the number of global iterations to be somewhat larger than in the case of full convergence of the subdomains. This is confirmed by Fig. 4, where the interlaced case is seen to require more iterations to converge in all time steps. On the other hand, this results in a more economical process since the number of internal iterations is greatly reduced.

The separation of the field and circuit domains facilitates the specification of initial conditions. The numerical differentiation in (6) has no negative impact on the stability scheme; in fact, the process remains stable for very large time steps.

VI. Conclusions

A scheme for coupling discretized field models of magnetic devices with circuit simulators has been presented, the motivation of which is to investigate the requirements, advantages and disadvantages of coupling separate solvers. Access to the domain Jacobians is required in order to construct the iteration matrix for the coupled problem. The

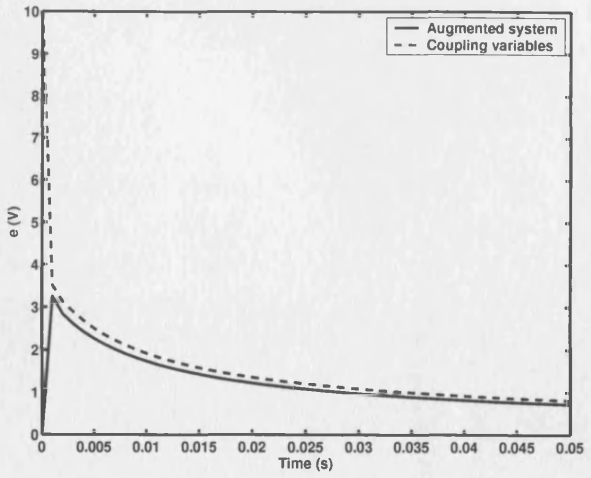


Fig. 2. Electromotive force in the inductor.

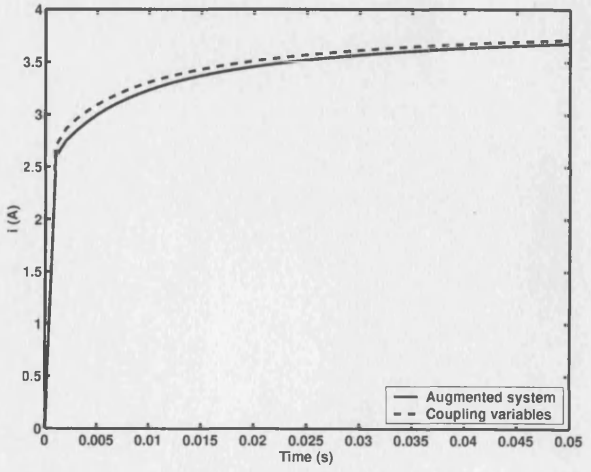


Fig. 3. Current in the inductor.

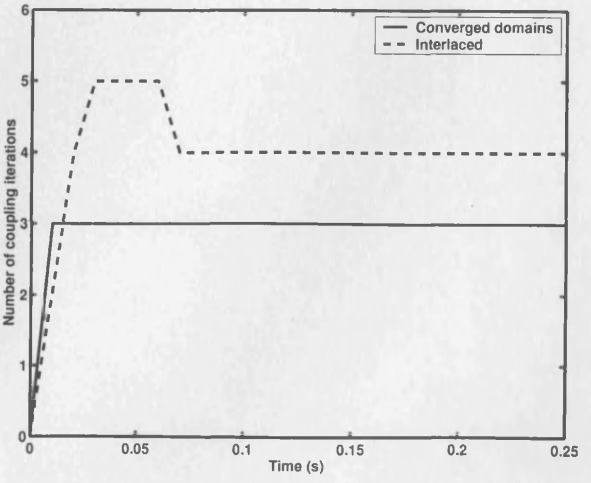


Fig. 4. Number of coupling iterations per time step.

implementation also requires a control program from where the domain solvers are called and the corresponding quantities retrieved. This scheme allows the use of existing circuit simulators and field solvers, but necessitates sensitivity calculations. The number of iteration of the nested solvers can be reduced by interlacing.

References

[1] J. Vaananen, "Circuit theoretical approach to couple two-dimensional finite element models with external circuit equations," *IEEE Transactions on Magnetics*, vol. 32, no. 2, pp. 400–410, March 1996.

[2] P. Eustache, G. Meunier, and J. L. Coulomb, "Finite element toolbox for generic coupling (magnetic, thermal,etc.)," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1461–1464, May 1996.

[3] N. R. Aluru and J. White, "A multilevel Newton method for mixed-energy domain simulation of MEMS," *Journal of Microelectromechanical Systems*, vol. 8, no. 3, pp. 299–307, September 1999.

[4] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits*, Prentice-Hall, Englewood Cliffs, 1975.

[5] D. N. Dyck, D. A. Lowther, and E. M. Freeman, "A method of computing the sensitivity of electromagnetic quantities to changes in materials and sources," *IEEE Transactions on Magnetics*, vol. 30, no. 5, pp. 3415–3418, September 1994.

[6] E. Melgoza, H. C. Lai, and D. Rodger, "Interlaced nonlinear iteration for coupled problems," in *CEFC'2000 Conference Proceedings*, Milwaukee, June 2000.